Iterative Power Algorithm for Global Optimization with Quantics Tensor Trains

Micheline B. Soley,‡† Paul Bergold,¶ and Victor S. Batista*,†‡

†Yale Quantum Institute, Yale University, P.O. Box 208334, New Haven, CT, 06520-8263, USA
‡Department of Chemistry, Yale University, P.O. Box 208107, New Haven, CT, 06520, USA
¶Zentrum Mathematik, Technische Universität München, Boltzmannstr. 3, 85748 Garching, Germany

E-mail: victor.batista@yale.edu
Abstract

We introduce the iterative power algorithm (IPA) for global optimization and a formal proof of convergence for both discrete and continuous global search problems. IPA implements the power iteration method in quantics tensor train (QTT) representations. Analogous to the imaginary time propagation method with infinite mass, IPA starts with an initial probability distribution \( \rho_0(r) \) and iteratively applies the recurrence relation

\[
\rho_{k+1}(r) = \frac{U \rho_k(r)}{\|U \rho_k(r)\|_{L^1}},
\]

where \( U = e^{-V(r)} \) is defined in terms of the ‘potential energy’ function \( V(r) \) with global minimum at \( r = r^* \). Upon convergence, the probability distribution becomes a delta function \( \delta(r - r^*) \), so the global minimum can be obtained as the position expectation value \( r^* = \text{Tr}[r \delta(r - r^*)] \). QTT representations of \( V(r) \) and \( \rho(r) \) are generated by fast adaptive interpolation of multidimensional arrays to bypass the curse of dimensionality and the need to evaluate \( V(r) \) for all possible values of \( r \). We illustrate the capabilities of IPA as applied to solving the factorization problem formulated as a global search optimization on the "potential energy" surface \( V(r) = \text{mod}(N, r) \), where \( N \) is the number to be factorized and \( r \in \{2, 3, 5, 7, 11, \ldots\} \) is the space of prime numbers folded as a \( d \)-dimensional \( 2_1 \times 2_2 \times \cdots \times 2_d \) tensor. We find that IPA resolves multiple degenerate global minima corresponding to prime factors of \( N \) even when separated by large energy barriers in the highly rugged landscape of \( V(r) \). Therefore, IPA should be of great interest for a wide range of other optimization problems ubiquitous in molecular and electronic structure calculations.


1 Introduction

The development of efficient optimization algorithms remains a subject of great research interest since optimization problems are central to important applications in many branches of science and engineering, including molecular and electronic structure calculations. In control theory, for example, global optimization algorithms are essential to determine the drives that steer a system into a desired final state. Another prototypical example is the problem of finding the minimum energy structure of a complex molecule, usually the first step in studies of molecular properties, molecular reactivity, and drug design. The simplest approach for finding the global optima in a discrete set is to sift through all possibilities. However, that approach becomes intractable for high dimensional systems since the number of possible states typically scales exponentially with the number of degrees of freedom — i.e., the so-called "curse of dimensionality" problem. Analogously, simple approaches for continuous optimization involve sampling stochastically or deterministically. Yet, these procedures typically lead to "trapping" in local minima. Therefore, the development of efficient global search algorithms remains an open problem of great interest.

In this paper, we build upon the strategy of the diffeomorphic modulation under observable-response-preserving homotopy (DMORPH) method and we introduce the iterative power algorithm (IPA) for global optimization. DMORPH evolves a distribution function $\rho(r)$ in the search space of configurations, so that the distribution becomes localized at the global optima and the minimum can be revealed by computing the position expectation value. Analogously, IPA implements the same strategy of evolving a probability distribution function although with a very different approach. Instead of implementing the DMORPH approach of iteratively optimizing control parameters of an externally applied field that localizes $\rho(r)$ at the global optimum, IPA applies a simple amplitude amplification scheme based on the power method. The resulting algorithm is essentially an imaginary time propagation although with infinite mass. The relation between the power method of linear algebra and the imaginary time propagation method has been previously discussed, although it
remains to be formally analyzed.

The power method is based on the recurrence relation $\rho_{k+1}(r) = U\rho_k(r)/\|U\rho_k(r)\|_{L1}$. In the IPA implementation, $U = e^{-V(r)}$ is defined by the scaled potential energy surface $V(r)$, and $\rho_k(r)$ is the density distribution after the $k$-th optimization step. Such an iterative procedure transforms any initial distribution with non-zero amplitude at the global minimum into a delta function $\rho(r) = \delta(r - r^*)$ (i.e., the eigenvector of $U(r)$ with maximum eigenvalue in the basis of Dirac delta functions). The global minimum can then be revealed, as in the DMORPH method, by computing the position expectation value $r^* = \text{Tr}[r\rho(r)]$.

IPA can efficiently find the global minimum of low-rank high-dimensional potential energy surfaces with $2^d$ possible states $r$ by approximating $\rho(r)$ and $V(r)$ in the form of quantics tensor trains (QTTs), a specific form of tensor trains (TT) or matrix product states (MPS) of great interest. The QTTs have arrays reshaped into $2 \times 2 \times \cdots \times 2$ tensors, so they represent high-dimensional quantities $Q(i_1, \ldots, i_d)$ with $2^d$ possible values. Since they depend on $d$ physical variables $i_k$ each of them with 2 possible values, they are decomposed into the outer product of tensor cores, as follows:

$$Q(i_1, \ldots, i_d) \approx \sum_{\alpha_1=1}^{r_1} \sum_{\alpha_2=1}^{r_2} \cdots \sum_{\alpha_{d-1}=1}^{r_{d-1}} A_1(1, i_1, \alpha_1)A_2(\alpha_1, i_2, \alpha_2) \cdots A_d(\alpha_{d-1}, i_d, 1),$$

where $Q$ is the reshaped $d$-dimensional tensor; $A_j$ are individual, order-three, rank $r_j$ tensor cores contracted over the auxiliary indices $\alpha_j$; and $i_1, \ldots, i_d \in \{1, 2\}$. The QTT format, introduced by Eq. (1), reduces the cost of evaluating $Q$ over the search space of $2^d$ possibilities to not more than $2^d r^2$ evaluations for the maximal rank $r = \max(r_1, \ldots, r_{d-1})$. In addition, quantics tensor trains feature the same exponential improvement in data sparsity given by quantum computers, which offers the possibility of developing methods like IPA that can be thought of as classical computing analogues of quantum computing algorithms.

Quantum search algorithms (e.g., the Grover’s search method) typically initialize a uniform superposition and evolve it multiple times until a measurement of the resulting
state can identify one out of $2^d$ possibilities with sufficiently high probability. Analogously, we initialize $\rho_0(\mathbf{r})$ as a uniform distribution in QTT format to enable sampling of the entire search space simultaneously. Iterative application of the recurrence relation amplifies the amplitude at the global minima, which yields a final density localized at the global minima. We prove that the number of steps required by IPA to amplify the amplitude of the global minimum to a probability higher than 50% scales logarithmically with the size of the search space, which provides a valuable global search methodology alternative to well-established optimization methods.\cite{69,71}

The paper is organized as follows. The IPA method is introduced in Sec. 2, followed by the analysis of convergence rate in Sec. 3 and a discussion in perspective of existing approaches in Sec. 4. Computational results are presented in Sec. 5, and conclusions in Sec. 6. Appendix A presents a formal proof of IPA convergence. Appendix B analyzes the convergence rate of the power method. Python codes to reproduce the reported calculations are provided in Appendices C and D.

2 The Iterative Power Algorithm Method

IPA solves the optimization problem

$$\min_{x \in \mathbb{R}^n} V(x), \quad (2)$$

for a given potential $V: \mathbb{R}^n \to \mathbb{R}$. Here, we limit the discussion to the one dimensional case $n = 1$ since any problem with $n > 1$ can be vectorized into an $n = 1$ version. To guarantee the existence of a global minimum,\cite{71} we assume $V(x)$ is continuous and coercive (i.e., $V(x) \to +\infty$ as $|x| \to +\infty$). Our goal is to compute the set of all minima locations of
\[
\begin{align*}
\arg\min_{x \in \mathbb{R}} V(x) &= \left\{ x^* \in \mathbb{R} \mid V(x) \geq V(x^*) \text{ for all } x \in \mathbb{R} \right\}. (3)
\end{align*}
\]

Therefore, we employ a non-negative probability density function \( \rho : \mathbb{R} \to [0, \infty) \) that is bounded and with unit norm:

\[
\|\rho\|_{L^1} = \int_{\mathbb{R}} dx \rho(x) = 1. (4)
\]

The initial density \( \rho_0 \) is supported (non-zero) around every minima \( x^* \) of the potential \( V \), so for all \( r > 0 \) the initial density \( \rho_0 \) satisfies the following condition:

\[
\int_{x^* - r}^{x^* + r} dx \rho_0(x) > 0. (5)
\]

In each IPA iteration, a transformation matrix \( U \) is applied from the left to \( \rho_0 \) to increase the density amplitude at the global minimum positions relative to amplitudes at the remainder of the search space. The resulting density distribution \( U\rho_0 \) is then normalized to obtain a new density \( \rho_1 \), which is the input for the next IPA iteration. Any \( U \) can be used, provided it satisfies the following two conditions: (i) \( U(x) \) must be a continuous and positive function that is maximized at the global minima of \( V \)

\[
\arg\max_{x \in \mathbb{R}} U(x) = \arg\min_{x \in \mathbb{R}} V(x), (6)
\]

and (ii) \( U(x) \) must be integrable (we denote this by \( U \in L^1(\mathbb{R}) \)).

A simple example is \( U(x) = e^{-\beta V(x)} \) for a fixed scaling parameter \( \beta > 0 \). We note that Eq. (6) holds for \( U(x) \) since the exponential is a strictly increasing function. Furthermore, the coercivity condition of the potential implies \( U(x) \) is integrable for a sufficiently fast growing potential \( V(x) \) in the asymptotic region \( |x| \to +\infty \).
2.1 Evolution: Amplitude Amplification

IPA generates a sequence of density distributions $\rho_1, \rho_2, \ldots$, starting from a uniform distribution $\rho_0$, as follows:

\begin{verbatim}
for k = 1, 2, \ldots
r_k = \|U\rho_{k-1}\|_{L^1} = \int_{\mathbb{R}} dx U(x) \rho_{k-1}(x);
\rho_k(x) = \frac{U(x)\rho_{k-1}(x)}{r_k} = \frac{U(x)^k \rho_0(x)}{\|U^k \rho_0\|_{L^1}};
\end{verbatim}

end

Since $U$ is assumed to be continuous and integrable, we conclude it is bounded and $L^1$-normalizable ($U \in L^\infty(\mathbb{R}) \cap L^1(\mathbb{R})$). In particular, this guarantees the normalization factors $r_k > 0$ are well-defined, since repeated applications of $U$ remain $L^1$-normalizable ($U^k \in L^1(\mathbb{R})$ for all iterations $k \geq 1$).

Appendix A proves the sequence $\rho_1, \rho_2, \ldots$ produced by IPA converges to a "Dirac comb"—i.e., a sum of Dirac delta functions located at the global minima $x_1^* < x_2^* < \cdots < x_m^*$ of the potential $V$ (which can be viewed as the limit of the so-called Dirac sequences, as mentioned in Appendix A):

$$\rho_{\text{final}}(x) = \lim_{k \to \infty} \rho_k(x) \to \sum_{j=1}^m \delta(x - x_j^*).$$

2.2 Resolution of Global Minima: Measurement

The global minima are obtained after obtaining $\rho_{\text{final}}(x)$, as follows:

(i) When $V(x)$ has a single global minimum at $x = x^*$, the minimum is obtained by computing the position expectation value with the final density $\rho_{\text{final}}(x)$:

$$x^* = \langle x \rangle_{\rho_{\text{final}}} = \int_{\mathbb{R}} dx x \rho_{\text{final}}(x).$$
(ii) When $V(x)$ has only two degenerate global minima (e.g., as for the factorization of biprimes discussed below), we first compute the position expectation value of $\rho_{\text{final}}$ to obtain the average position of the two global minima $\bar{x}$. Then, we multiply $\rho_{\text{final}}$ by a Heaviside step function,

$$\Theta(x-\bar{x}) = \begin{cases} 
0, & \text{if } x \leq \bar{x}, \\
1, & \text{if } x > \bar{x},
\end{cases}$$

(9)

to obtain the distributions $\rho_{\text{final}}(x)\Theta(x-\bar{x})$ and $\rho_{\text{final}}(x)(1-\Theta(x-\bar{x}))$, which are single delta functions resolving the two minima.

(iii) When $V(x)$ has an unknown number of global minima, we first obtain $\rho_{\text{final}}$. Then, we reinitialize $\rho_0 = \rho_{\text{final}}$ and we run IPA with a "ramp potential" rather than using the potential of the problem of interest. The ramp is usually a simple monotonically increasing function (e.g., $V(x) = x$) that breaks the degeneracy of the Dirac comb by amplifying the amplitude of the minimum of all minima (i.e., $x^*_1$). After computing $x^*_1$ as an expectation value, we multiply $\rho_{\text{final}}$ by the Heaviside function $\Theta(x-x^*_1)$ introduced by Eq. (9) and we repeat the IPA ramp to identify the second minima. The scheme is then repeated until all global minima are resolved.

2.3 QTT Representation

IPA is not limited to a specific choice of basis set representation for $\rho(x)$, $V(x)$ and $U(x)$. However, we employ the Quantics Tensor Train (QTT) representation\textsuperscript{61-63} generated by fast adaptive interpolation of multidimensional arrays as implemented in Oseledets’ TT-Toolbox.\textsuperscript{73} The resulting implementation bypasses the curse of dimensionality, which allows for applications to high dimensional potentials (Python scripts provided in Appendices C and D).
3 Convergence Rate Analysis

Appendix A provides a formal proof of convergence for IPA continuous global optimization. Here, we focus on discrete optimization for a problem with a single global minimum. We show that the number of IPA steps necessary to amplify the amplitude of the global minimum to a value higher than $1/2$ scales logarithmically with the number $N$ of possible states. The analysis is analogous to the estimation of the number queries required for amplitude amplification by Grover’s algorithm.

First, we show that IPA converges to the global minimum for the specific case where $U$ is given by an $N \times N$ diagonal matrix $U$ with $N \geq 1$ positive entries (eigenvalues $\lambda_j$ with $j = 1, \cdots, N$) with a unique maximum $\lambda_1 > 0$. For simplicity, we take all other eigenvalues to be $\lambda_2$, with

$$0 < \lambda_2 < \lambda_1. \quad (10)$$

Hence, $U$ can be expressed as follows:

$$U = \text{diag} (\lambda_2, \ldots, \lambda_2, \lambda_1, \lambda_2, \ldots, \lambda_2) \in \mathbb{R}^{N \times N}, \quad (11)$$

where $\lambda_1$ is the $k$-th diagonal entry for some $1 \leq k \leq N$. An illustration of $U$ is given in Fig. [1].

We consider an initial density given by the discrete uniform distribution

$$\rho_0 = \frac{1}{N} (1, \ldots, 1) \in \mathbb{R}^N. \quad (12)$$

The $k$-th IPA iteration updates the density distribution, as follows:

$$\rho_k = \frac{U \rho_{k-1}}{\|U \rho_{k-1}\|_1} = \frac{U^k \rho_0}{\|U^k \rho_0\|_1}, \quad (13)$$

$$= \frac{u_k}{\|u_k\|_1}, \quad (14)$$
Figure 1: Illustration of the vector form of $U$ assumed to have a unique maximum eigenvalue $\lambda_1$ and all other eigenvalues of equal amplitude $\lambda_2$.

where repeated application of $U$ yields:

$$u_k = (\lambda_2^k, \ldots, \lambda_2^k, \lambda_1^k, \lambda_2^k, \ldots, \lambda_2^k), \quad (15)$$

with norm

$$\|u_k\|_1 = \sum_{j=1}^{N} |(u_k)_j| = \lambda_1^k + (N-1)\lambda_2^k. \quad (16)$$

We note that $\lambda_1^k > \lambda_2^k$ since $\lambda_1 > \lambda_2$, so the vector $\rho_k$ produced after $k$ iterations has $N \geq 1$ positive entries, a unique maximum

$$\rho_{k,\text{max}} = \max_{j=1,\ldots,N} (\rho_k)_j = \frac{\lambda_1^k}{\|u_k\|_1}, \quad (17)$$

and all other entries with value

$$\rho_{k,\text{min}} = \min_{j=1,\ldots,N} (\rho_k)_j = \frac{\lambda_2^k}{\|u_k\|_1}. \quad (18)$$
Therefore, the minimum to maximum amplitude ratio is

\[
\frac{\rho_{k,\text{min}}}{\rho_{k,\text{max}}} = \left(\frac{\lambda_2}{\lambda_1}\right)^k.
\]

(19)

Each IPA iteration decreases the ratio by a factor of \(\lambda_2/\lambda_1 < 1\) while the norm is conserved. Therefore, only the maximum entry of the state vector \(\rho_k\) survives in the limit of an infinite number of iterations \(k \to +\infty\).

Using the normalization condition,

\[
1 = \|\rho_k\| = \rho_{k,\text{max}} + (N - 1)\rho_{k,\text{min}},
\]

(20)

and inserting the ratio given by Eq. (19) into the normalization condition introduced by Eq. (20), we can solve for the maximum amplitude \(\rho_{k,\text{max}}\), as follows:

\[
\rho_{k,\text{max}} = \frac{1}{1 + (N - 1) \cdot (\lambda_2/\lambda_1)^k},
\]

(21)

which converges to 1 in the limit \(k \to \infty\).

The number of iterations required to amplify the amplitude of the global minimum to a value higher than or equal to \(1/2\) is

\[
\frac{1}{1 + (N - 1) \cdot (\lambda_2/\lambda_1)^k} \geq \frac{1}{2}.
\]

(22)

Solving this inequality gives the minimum number of required IPA iterations,

\[
k \geq \frac{\log (N - 1)}{\log (\lambda_1/\lambda_2)},
\]

(23)

which scales logarithmically with the size of the search space \(N\) and inverse logarithmically with the ratio of eigenvalues \(\lambda_1/\lambda_2\).
4 Comparison to Other Methods

IPA can be compared to the power method and imaginary time propagation. The connection between the power method and imaginary time propagation has been discussed, although the relationship between the two methods has yet to be formally analyzed.

We begin with the recurrence relation of the power method. For a matrix $U \in \mathbb{C}^{N \times N}$ with eigenvalues $\lambda_1, \ldots, \lambda_N \in \mathbb{C}$, the subscripts denote the order $|\lambda_1| > |\lambda_2| \geq \cdots \geq |\lambda_N|$. Given a starting vector $\rho_0 \in \mathbb{C}^N$ that has a non-zero amplitude along the direction of the eigenvector with largest eigenvalue $\lambda_1$, the power method produces the following sequence of vectors $\rho_k \in \mathbb{C}^N$:

$$
\rho_k = \frac{U \rho_{k-1}}{\|U \rho_{k-1}\|} = \frac{U^k \rho_0}{\|U^k \rho_0\|},
$$

a sequence that converges to an eigenvector associated with the largest eigenvalue $\lambda_1$ independently of the norm $\| \cdot \|$. The resulting convergence is geometric in the ratio,

$$
\left| \frac{\lambda_2}{\lambda_1} \right| < 1.
$$

We note that according to the recurrence relation, introduced by Eq. (24), imaginary time propagation is essentially the power method where $\rho_0$ represents a trial initial wavefunction in a given basis set and $U$ is the matrix representation of the Boltzmann operator $e^{-\beta \hat{H}}$, where the Hamiltonian $\hat{H}$ is typically $\hat{H} = \hat{p}^2/(2m) + V(x)$ with $m$ the mass and $\hat{p}$ the momentum operator.

In IPA, however, $\rho_0$ is a probability density and $U$ can be any integrable, continuous, and positive function of $x$ that is maximal at the global minimum of $V$. As a result, IPA finds the global minima of $V(x)$ while the imaginary time propagation method finds the eigenstate of the Hamiltonian with minimum eigenvalue (i.e., the ground state). For the particular choice of $U(x) = e^{-\beta V(x)}$, however, IPA corresponds to imaginary time propagation with $m = \infty$. 
Eq. (24) also shows that IPA differs from the power method because it employs $U \in L^\infty(\mathbb{R}) \cap L^1(\mathbb{R})$ that meets the conditions described in Section 2 and a probability density function $\rho_0 \in L^1(\mathbb{R})$ to find the global minima, whereas the power method employs an arbitrary matrix $U \in \mathbb{C}^{N \times N}$ and a discrete vector $\rho_0 \in \mathbb{C}^N$ to find an eigenvector. This relationship also allows us to use the power method to analyze the convergence rate of IPA for discrete problems, as discussed in Appendix B.

5 Computational Results

This section shows that IPA successfully finds the degenerate global minima $r_1^* < r_2^* < \ldots < r_m^*$ of the potential,

$$V(r) = N \mod r,$$

(26)

corresponding to the prime factors of $N$, when formulating the factorization problem as a rather challenging global minimum energy search. The modulo operation that defines $V(r)$ in the space of prime numbers $r$ returns the remainder after division of $N$ by $r$. To factorize large numbers, the Python scripts provided in Appendices C and D represent $N$ and operations on $N$ with 3000-digit precision, using the mpmath library, which shows that IPA can resolve multiple degenerate prime factors of a large number $N$ with thousands of digits of the form,

$$N = (r_1^*)^{e_1} \times (r_2^*)^{e_2} \times \cdots \times (r_m^*)^{e_m},$$

(27)

where $e_j \geq 1$ is the degeneracy of the prime factor $r_j$. A simple example for $N = 187$ is shown in Fig. 2 where the global optima are $r_1^* = 11$ and $r_2^* = 17$. 
Figure 2: Scaled potential energy surface ($\log(1+V(r))$) for optimization of $V(r) = N \mod r$, with $N = 187$. The global minima (starred) correspond to the prime factors of $N = 11 \times 17$.

The Python script provided in Appendix C successfully resolves multiple degenerate global minima, regardless of the number of minima, their degeneracy, the distance between minima, or the potential energy barrier separating the minima.

The QTT approximation of $\rho_0(r)$ provides an accurate and efficient representation of the initial uniform distribution in the search space of prime numbers $r \in \{2, 3, 5, 7, 11, \ldots\}$, folded as a $d$-dimensional $2_1 \times 2_2 \times \cdots \times 2_d$ tensor. The distribution evolves according to the IPA recurrence relation, which increases amplitude at the global optima while reducing it elsewhere. Application of $U = e^{-\beta V}$ with scaling parameter $\beta = 30$ yields a numerically converged final density in only three IPA iterations.

Figure 3 shows that IPA correctly amplifies the amplitude of the degenerate prime factors of $N = (3^2 \times 11 \times 17 \times 23 \times 41 \times 53 \times 79 \times 101 \times 109)^{200}$, a large integer with 2,773 digits (more than 9,212 bits). Consistent with a Dirac sequence, the final density is maximal for the prime factors of $N$ and nearly zero elsewhere in the search space of primes. Measurement with the ramp function, as described in Sec. 2.2, then successfully resolves the individual prime factors as shown in Fig. 4. IPA thus correctly determined the position of all global
minima of the test potential function.

Figure 3: As expected, the IPA procedure for global optimization of the function Eq. (26), for $N$ as defined in the text, yielded a final density in the form of a Dirac comb that was maximal at positions of global optima and zero elsewhere. Only a fraction of the search space of prime numbers is illustrated for clarity.
Figure 4: The Dirac delta components of the final density in IPA were successfully isolated without evaluation of the function at all points on the search space via the ramp method for $U = e^{-\tilde{\beta} \text{ramp}}$ with parameter $\tilde{\beta} = 0.5$ and found to be located at the global optima of the function Eq. (26) for the large number $N$. Given the size of the search space of prime numbers, the density is shown in a restricted region to enable visualization of its maximal values.

Figure 5 shows the IPA execution time as a function of $N$ when solving the factorization problem of biprimes $N = p_1 \times p_2$, where $p_1$ and $p_2$ are primes with values up to 9998000099 and where $U = e^{-\beta V}$ with $\beta = 20$ (requiring only one IPA iteration). The regression analysis shows that the execution time scales approximately as $O(\ln(N))$ ($R^2 = 0.978$), or $O(\ln(\ln(N)))$ ($R^2 = 0.977$). The logarithmic scaling agrees with the analysis of Section 3, which shows that the resulting scaling for amplitude amplification is comparable to or better than in optimal quantum search algorithms (e.g., the Grover quantum search method, where the number of queries necessary to amplify the amplitude of one out of $N$ possible states scales as $O(\sqrt{N})$).
6 Discussion

The QTT implementation of IPA illustrates the possibility of developing efficient algorithms for classical computing. Analogous to quantum computing algorithms, superposition states can be evolved by applying a sequence of unitary transformations, and the outcome of the calculation corresponds to a "measurement" (i.e., an expectation value obtained with the evolved superposition). The QTT representation avoids the curse of dimensionality, enabling benchmark calculations that would be otherwise impossible on classical high-performance computing facilities. We find that such a computational strategy enables IPA to perform quite efficiently, bypassing the usual limitations of traditional optimization methods. Therefore, it is natural to anticipate that IPA should be of great interest for a wide range of applications, including optimization problems in molecular and electronic structure calculations.
Acknowledgements

The authors are grateful for conversations with Dr. Caroline Lasser, Dr. Erik T. J. Nibbering, and Dr. Maximilian Engel. M. B. S. acknowledges financial support from the Yale Quantum Institute Postdoctoral Fellowship, the National Science Foundation Graduate Research Fellowship grant number DGE-1144152 and the Blue Waters Graduate Research Fellowship, part of the Blue Waters sustained-petascale computing project supported by the National Science Foundation (awards OCI-0725070 and ACI-1238993) and the State of Illinois. Blue Waters is a joint effort of the University of Illinois at Urbana-Champaign and its National Center for Supercomputing Applications. V.S.B. acknowledges support from the NSF Grant No. CHE-1900160, and high performance computing time from NERSC and the Yale High Performance Computing Center.

TOC Graphic
Appendix

A Proof of Convergence

This section shows that the sequence generated by the IPA recurrence relation converges to a delta distribution \( \delta(x - x^*) \) when \( V(x) \) has a single global minimum at \( x = x^* \). An analogous proof can be provided for surfaces with multiple global minima by generalization of the concept of a Dirac sequence.

The sequence of densities \( \rho_k \) converges to the delta distribution as the Dirac sequence:

(i) For all \( k \in \mathbb{N} \) and all \( x \in \mathbb{R} \): \( \rho_k(x) \geq 0 \),

(ii) For all \( k \in \mathbb{N} \): \( \rho_k \in L^1(\mathbb{R}) \) and \( \int_{\mathbb{R}} dx \rho_k(x) = 1 \),

(iii) For all \( \varepsilon > 0 \): \( \lim_{k \to \infty} \int_{\mathbb{R} \setminus (x^* - \varepsilon, x^* + \varepsilon)} dx \rho_k(x) = 0 \), where the integral is evaluated over the real line \( \mathbb{R} \) except the interval \( (x^* - \varepsilon, x^* + \varepsilon) \).

These conditions guarantee the area under the curve \( \rho_k \) is concentrated near the global minimum location \( x^* \), provided the number of iterations \( k \) is sufficiently large.

The properties (i) and (ii) follow by construction of the IPA sequence. To prove property (iii), let \( \varepsilon > 0 \) be a positive distance. For a radius \( r > 0 \), we denote the minimum of \( U \) on the interval \( [x^* - r, x^* + r] \) by

\[
m_r = \min_{x \in [x^* - r, x^* + r]} U(x).
\]
Since $U$ is continuous with a single global maximum at $x^*$, there exists a radius $r_\varepsilon > 0$ such that the number $m_{r_\varepsilon}$ is a positive and strict upper bound for $U$ outside the interval $(x^* - \varepsilon, x^* + \varepsilon)$, as follows (Figure 6):

$$\frac{U(x)}{m_{r_\varepsilon}} < 1, \quad \text{for all } x \in \mathbb{R} \setminus (x^* - \varepsilon, x^* + \varepsilon).$$  \hspace{1cm} (29)$$

Figure 6: There exists a radius $r_\varepsilon > 0$ such that the minimum $m_{r_\varepsilon}$ on $[x^* - r_\varepsilon, x^* + r_\varepsilon]$ is a strict upper bound (orange) for all values outside the interval $(x^* - \varepsilon, x^* + \varepsilon)$ (shown in blue), since $U$ is a continuous function with a single global maximum at $x = x^*$.

We then introduce the probability,

$$p_\varepsilon = \int_{x^* - r_\varepsilon}^{x^* + r_\varepsilon} dx \rho_0(x) > 0,$$  \hspace{1cm} (30)$$

and according to the definition of the minimum $m_{r_\varepsilon} > 0$, introduced by Eq. (28), for all $k \geq 1$ we obtain the norm,

$$\|U^k \rho_0\|_{L^1} = \int_{\mathbb{R}} dx \, U(x)^k \rho_0(x) \geq m_{r_\varepsilon}^k \int_{x^* - r_\varepsilon}^{x^* + r_\varepsilon} dx \, \rho_0(x) = m_{r_\varepsilon}^k p_\varepsilon,$$  \hspace{1cm} (31)$$
which gives the bound,

\[ \rho_k(x) = \frac{U(x)^k \rho_0(x)}{\|U^k \rho_0\|_{L^1}} \leq \|\rho_0\|_{\infty} \left( \frac{U(x)}{m_{r\varepsilon}} \right)^k \] for all \( x \in \mathbb{R} \), \hspace{1cm} (32)

where \( \|\rho_0\|_{\infty} \) is the supremum \( \sup_{x \in \mathbb{R}} |\rho_0(x)| \). According to Eq. (29), \( \frac{U(x)}{m_{r\varepsilon}} < 1 \) for all positions outside the interval \((x^* - \varepsilon, x^* + \varepsilon)\). Hence, we conclude that the density after \( k \) iterations is bounded for all those positions \( x \) and all iterations \( k \geq 1 \), as follows:

\[ \rho_k(x) \leq \|\rho_0\|_{\infty} \frac{U(x)}{p_{\varepsilon} m_{r\varepsilon}}, \] \hspace{1cm} (33)

showing that the sequence is dominated by an integrable function. Thus, the Lebesgue dominated convergence theorem yields

\[ \lim_{k \to \infty} \int_{\mathbb{R} \setminus (x^* - \varepsilon, x^* + \varepsilon)} dx \rho_k(x) = \int_{\mathbb{R} \setminus (x^* - \varepsilon, x^* + \varepsilon)} dx \lim_{k \to \infty} \rho_k(x) = 0. \] \hspace{1cm} (34)

\section{B Power Method: Convergence Rate Analysis}

We consider a diagonal matrix \( U \in \mathbb{R}^{N \times N} \) whose entries are given by the value of \( U \) at the equally spaced positions \( x_1, \ldots, x_N \) with \( \Delta x = x_{j+1} - x_j = (b - a)/N \) in the finite interval \( x = [a, b] \),

\[ U = \text{diag} \left( U(x_1), U(x_2), \ldots, U(x_N) \right). \] \hspace{1cm} (35)

We consider an initial vector whose entries are given by the value of the initial density \( \rho_0 \) at the same positions,

\[ \rho_0 = (\rho_0(x_1), \rho_0(x_2), \ldots, \rho_0(x_N)) \in \mathbb{R}^N. \] \hspace{1cm} (36)
When $N$ is sufficiently large, we obtain the following approximation for all iterations:

$$\|U^k \rho_0\|_1 = \sum_{j=1}^{N} U(x_j)^k \rho_0(x_j) \approx \frac{1}{\Delta x} \int_{\mathbb{R}} dx U(x)^k \rho_0(x) = \frac{1}{\Delta x} \|U^k \rho_0\|_{L^1}. \quad (37)$$

In the following, we denote by $\rho^* \in \mathbb{R}^N$ the vector whose $j$-th coordinate equals one if $U(x_j) = \lambda_1$ is the dominant eigenvalue of $U$ and zero otherwise. Moreover, we introduce the constant,

$$c = \frac{1}{\# \{j \mid U(x_j) = \lambda_1\}}, \quad (38)$$

where we use the notation $\# A$ for the cardinality (i.e., number of elements in the set). The definition of $U$ in Eq. (35) yields that the sequence $\rho_1, \rho_2, \ldots$ produced by the power iteration (i.e., Eq. (24) using the norm $\| \cdot \|_1$) converges to $c \cdot \rho^*$. Using the approximation in Eq. (37), we conclude that the density $\rho_k$ produced by IPA can be approximated at a given grid point $x_j$ as

$$\rho_k(x_j) = \frac{U(x_j)^k \rho_0(x_j)}{\|U^k \rho_0\|_{L^1}} \approx \frac{1}{\Delta x} \frac{(U^k \rho_0)_j}{\|U^k \rho_0\|_1} \rightarrow \frac{c}{\Delta x} \delta_{j,n}. \quad (39)$$

In the special case where $U$ has a unique dominant eigenvalue, say $\lambda_1 = U(x_n)$ for some unique $n \in \{1, \ldots, N\}$, we get $\rho^*_j$ is the Kronecker delta $\delta_{j,n}$. This allows us to confirm IPA generates a Dirac sequence at the global minimum for discrete optimization problems. The relationship of this expression to that of the power method also shows IPA inherits the geometric convergence rate in the ratio $\lambda_2/\lambda_1 < 1$ from the power method, in agreement with the alternative analysis introduced in Section 3.

To further specify the convergence rate of IPA, we relate the ratio $\lambda_2/\lambda_1$ of the power method to the spacing $\Delta x$ in IPA. This is accomplished by classifying the steepness of $U$ around its maximum location $x^*$ via local approximations by polynomials of even degree. If there exists a positive parameter $\alpha > 0$ and an integer $m \geq 1$ such that, for all positions
within a distance of $\Delta x$ of the maximum $x^*$, $U$ is bounded from below by

$$U(x) \geq U(x^*) - \alpha (x - x^*)^{2m}, \quad (40)$$

then the eigenvalue $\lambda_2$ is bounded from below by $U(x^*) - \alpha \Delta x^{2m}$. Therefore, we conclude the rate of convergence is bounded as

$$\frac{\lambda_2}{\lambda_1} \geq \frac{U(x^*) - \alpha \Delta x^{2m}}{U(x^*)} = 1 - \frac{\alpha}{U(x^*)} \Delta x^{2m}. \quad (41)$$

In particular, $\lambda_2/\lambda_1 \to 1$ as $\Delta x \to 0$.

C Multiple Degenerate Global Minima

The following Python script illustrates the implementation of IPA as applied to finding multiple degenerate global minima corresponding to the degenerate prime factors of the integer $N = (3^2 \times 11 \times 17 \times 23 \times 41 \times 53 \times 79 \times 101 \times 109)^{200}$ with 2,773 digits, when using the ttpy library installed from [http://github.com/oseledets/ttpy](http://github.com/oseledets/ttpy).

```python
import numpy as np
from numpy import zeros, reshape, sqrt, arange, vectorize, extract, int
import matplotlib.pyplot as plt
import tt
import mpmath
from mpmath import mpf, floor, exp, nint

def parameters():
    global dim, eps, num, rmax, nsteps, d, searchspacesize, beta, betaprime
    num=mpf(3*3*11*17*23*41*53*79*101*109)**200
    beta=30
    betaprime=0.5
    dim=1
    eps=1.0e-100
    rmax=100
    nsteps=3
    d=6
```

23
searchspacesize=2**d
return ()

def rho0(input):
    V=1.0+0*input
    return V

def is_prime(n):
    if n % 2 == 0 and n > 1:
        return False
    return all(n % i for i in range(3,int(sqrt(n))+1,2))

def tto(input, param=None):
    global num,beta
    nevals,dim=input.shape
    out=np.zeros((nevals,))
    for ii in range(nevals):
        a=num-nint(input[ii,0])*floor(num/nint(input[ii,0]))
        out[ii]=input[ii,1]*exp(-beta*a)
    return(out)

def newtto(input, param=None):
    global num,betaprime
    nevals,dim=input.shape
    out=np.zeros((nevals,))
    for ii in range(nevals):
        a=input[ii,0]
        out[ii]=input[ii,1]*exp(-betaprime*a)
    return(out)

def ttround(input, param=None):
    nevals,dim=input.shape
    out=np.zeros((nevals,))
    for ii in range(nevals):
        out[ii]=np.round(input[ii,1])
    return(out)

def ttremovefactor(input, param=None):
    global largestfactor
    nevals,dim=input.shape
    out=np.zeros((nevals,))
    for ii in range(nevals):
        if input[ii,0]-0.5 < largestfactor:
            out[ii]=0.
        else:
            out[ii]=np.round(input[ii,1])
    return(out)
if __name__ == "__main__":
    global eps, num, rmax, nsteps, d, searchspacesize, largestfactor
    np.random.seed(1234)
    mp.dps = 3000
    parameters()
    print(num)
    a = arange(2, 10 ** 6)
    foo = vectorize(is_prime)
    pbools = foo(a)
    primes = extract(pbools, a)
    pp = np.zeros(searchspacesize, dtype=float)
    for j in range(searchspacesize):
        pp[j] = primes[j]
    ttp = tt.tensor(reshape(pp, [2] * d))
    lprimes = []
    ttrho = tt.multifuncrs([ttpp], rhoo, eps, verb=0, rmax=rmax)
    for k in range(nsteps):
        ttrho = tt.multifuncrs([ttpp, ttrho], tto, eps, verb=0,
                               rmax=rmax)
    ttrhostore = tt.multifuncrs([ttpp, ttrho], ttround, eps, verb=0,
                                rmax=rmax)
    ttrho = ttrho * (1.0 / (ttrho.norm()) ** 2)
    plt.bar(pp, reshape(ttrho.full(), searchspacesize), color='red',
            ls='-', label='ttrho')
    plt.xlabel("Prime Number")
    plt.ylabel("Density [arb. units.]")
    plt.title("Optimized Density")
    plt.xlim(1, 128)
    plt.pause(1.0)
    plt.savefig('diraccomb.png')

    largestfactor = 0.
    count = 0
    while num > 1:
        count = count + 1
        ttrho = tt.multifuncrs([ttpp, ttrhostore], ttremovefactor, eps,
                               verb=0, rmax=rmax)
        for k in range(nsteps):
            ttrho = tt.multifuncrs([ttpp, ttrho], newtto, eps, verb=0,
                                   rmax=rmax)
        ttrho = ttrho * (1.0 / ttrho.norm())
        ev = nint(tt.dot(ttpp, ttrho))
        largestfactor = ev
        lprimes.append(ev)
        num = num / ev
        while nint(num) % nint(ev) == 0:
            lprimes.append(ev)
num=num/ev
plt.clf()
plt.bar(pp, reshape(ttrho.full(), searchspacesize), color='red', ls='-',
       label='Density')
plt.xlabel("Prime Number")
plt.ylabel("Density [arb. units."]
plt.title("Prime Factor %i" % count)
print("prime factors=",lprimes,num)
plt.xlim(1,128)
plt.savefig('primefactor'+str(count)+'.png')

D Paired Degenerate Global Minima

The following Python script illustrates the implementation of IPA as applied to finding the
prime factors of the biprime $N = 99989 \times 99991$ by resolving the degenerate global minima
of the mod function, as described in the text, while using the ttpy library installed from

import numpy as np
from numpy import zeros, reshape, sqrt, arange, vectorize, extract, int, empty_like
import tt
import mpmath
from mpmath import mp, mpf, floor, exp, nint

def parameters():
    global dim, eps, num, rmax, nsteps, d, searchspacesize, beta
    num=mpf(99989*99991)
sqrtnum=sqrt(num)
    if sqrtnum < 24:
        d=3
    elif sqrtnum < 60:
        d=4
    elif sqrtnum < 138:
        d=5
    elif sqrtnum < 314:
        d=6
    elif sqrtnum < 728:
        d=7
    elif sqrtnum < 1622:
        d=8
    elif sqrtnum < 3674:
        d=9
elif sqrt(num) < 8168:
    d = 10
elif sqrt(num) < 17882:
    d = 11
elif sqrt(num) < 38892:
    d = 12
elif sqrt(num) < 84048:
    d = 13
elif sqrt(num) < 180512:
    d = 14
else:
    print("Error: Dimension not implemented.")
    quit()

rmax = 100
beta = 20
dim = 1
eps = 1.0e-100
nsteps = 1
searchspacesize = 2**d
return ()

def rhoo(input):
    V = 1.0 + 0 * input
    return V

def is_prime(n):
    if n % 2 == 0 and n > 1:
        return False
    return all(n % i for i in range(3, int(sqrt(n)) + 1, 2))

def tto(input, param=None):
    global num, beta
    nevals, dim = input.shape
    out = np.zeros((nevals,))
    for ii in range(nevals):
        a = num - nint(input[ii, 0]) * floor(num / nint(input[ii, 0]))
        if a > 10:
            a = 10
        out[ii] = input[ii, 1] * exp(-beta * a)
    return out

if __name__ == "__main__":
    global eps, num, rmax, nsteps, d, searchspacesize, ttavg
    np.random.seed(1234)
    mp.dps = 2000
    parameters()
    print(num)
a=arange(2, 10**6)
foo=vectorize (is_prime)
pbools=foo(a)
primes=extract(pbools, a)
pp=np.zeros(searchspacesize, dtype=float)
for j in range(searchspacesize):
    pp[j]=primes[j]
ttpp=tt.tensor(reshape(pp, [2]*d))
lprimes=[]
ttrho=tt.multifuncr2([ttpp], rhoo, eps, verb=0, rmax=rmax)
for k in range(nsteps):
    ttrho=tt.multifuncr([ttpp, ttrho], tto, eps, verb=0, rmax=rmax)
    ttrho=ttrho*(1.0/(ttrho.norm())**2)
ttrhostore=ttrho
ttavg=nint(tt.dot(ttpp, ttrho))
heaviside=empty_like(pp)
for j in range(searchspacesize):
    if pp[j]-0.5 > ttavg:
        heaviside[j]=0.
    else:
        heaviside[j]=1.
ttheaviside=tt.tensor(reshape(heaviside, [2]*d))
ttrho=ttheaviside*ttrhostore
ev=nint(tt.dot(ttpp, ttrho*(1.0/(ttrho.norm()))))
um=nint(num/ev)
lprimes.append(ev)
lprimes.append(num)
print("prime factors=", lprimes, num)

References


(62) Khoromskij, B. N.; Oseledets, I. V. DMRG+QTT approach to computation of the ground state for the molecular Schrödinger operator. 2010,


(73) Oseledets, I. oseledets/TT-Toolbox. 2020; [https://www.github.com/oseledets/TT-Toolbox](https://www.github.com/oseledets/TT-Toolbox)