One-dimensional Schrödinger equation:

Fast and Precise

Dmitry Levkov
INR RAS & ITMP MSU

Quarks-2026
Petrozavodsk, Russia

Sergei Demidov, DL, *to be published*

One-dimensional Schrödinger equation

$$\hat{H}\Psi \equiv -\frac{1}{2}\partial_x^2\Psi + V(x)\Psi = E\Psi$$

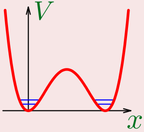
$$\boxed{\hbar = m = 1}$$

- Zillion numerical methods
- Automatic packages

} What else do we need?

- **Solve precisely? (100—1000 digits)** ← playground for precision calc

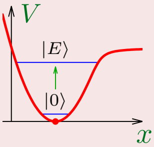
Tunneling & Instantons



$$\Delta E \propto \underbrace{e^{-F/g^2}}_{\text{exp. small}}$$

$g^2 \ll 1$ — semiclassical constant

Matrix elements



$$\mathcal{P} \propto |\langle E|\hat{x}|0\rangle|^2 \sim \underbrace{e^{-2\mathcal{W}_s/g^2}}_{\text{WF overlap}}$$

High-order perturbative series

$$\langle 0|\hat{B}|0\rangle = \sum B_n\, g^{2n} + \underbrace{\mathcal{O}(g^{2N})}_{\text{small}}$$

Summation methods

$$\langle 0|\hat{B}|0\rangle = \underbrace{\text{Borel}}_{\text{exp. small corrections}}\left[\sum B_n\, g^{2n}\right]$$

Need reliable testing methods!

Schrödinger equation — precisely

Few methods provide multiprecision (100-1000 digits)

- **Specific potentials:** realistically, $V = \sum V_n\, x^n$ $\boxed{\text{4th or 6th order}}$
  - Taylor series: $\Psi = \sum C_n x^n$
  - Riccati-Pade method: $\Psi = \dfrac{\sum C_n x^n}{\sum C'_m x^m}$

  $\Big\}$ recurrent relations for C_n, C'_m

  Fast, but efficient only for polynomial $V(x)$

- **Rayleigh-Ritz methods:** $\Psi = \sum\limits_{k=0}^{N} \psi_k \cdot \Phi_k(x)$ ← basis functions

  $[+]$ $\psi_N \sim \mathrm{e}^{-c\cdot N^\alpha}$ $\Rightarrow$ small cutoff error $[+]$ Generic potentials

$\boxed{\Phi_k =}$

- Oscillator WFs *(Hill '1995)*
- Orthogonal polynomials $P_k(x)$
  Lagrange mesh method *(e.g. Baye '2015)*
- **Preudospectral methods** $\sin(p_k x)$
- Sinc collocation method $\mathrm{sinc}[(x - a_k)/h]$
  (e.g. Fernández, Garcia '2015)
- ...

$\Big\}$ Schrödinger Eq.

$$\sum_{k'} H_{kk'}\, \psi_{k'} = E\psi_k$$

$N \times N$ eigensystem!

Rayleigh-Ritz methods are time-consuming!

$$\boxed{\sum_{k'} H_{kk'}\,\psi_{k'} = E\,\psi_k}$$

[$-$] Long calculation of $H_{kk'} = \underbrace{\langle\Phi_k|\hat{H}|\Phi_{k'}\rangle}_{N\times N \text{ integrals}}$ ← with multiprecision!

Ways out:

- [$+$] Pseudospectral methods — Fast Fourier Transform (FFT)
- [$+$] Lagrange mesh method — Special choice of polynomials

[$-$] Long solution of $\underbrace{N \times N \text{ system!}}_{N^3 \text{ operations}}$

- [$-$] Ways out: only for specific potentials

Need General, Fast & Precise method!

Key idea: combine Rayleigh-Ritz and lattice methods!

Newton iterations: improve precision

- Start with approximation: $\Psi^{(0)}$, $E^{(0)}$
- Find corrections: $\Psi = \Psi^{(0)} + \delta\Psi$, $E = E^{(0)} + \delta E$

$$\Rightarrow \boxed{(\hat{H} - E^{(0)})\delta\Psi - \delta E\,\Psi^{(0)} = (\hat{H} - E^{(0)})\Psi^{(0)}}$$

linearized Schrödinger equation

- Update approximation: $\Psi^{(0)} \to \Psi^{(0)} + \delta\Psi$
  $E^{(0)} \to E^{(0)} + \delta E$

Bonus: poor–precision iterations!

$$\underbrace{(\hat{H} - E^{(0)})\delta\Psi - \delta E\,\Psi^{(0)}}_{\substack{\text{lattice} \Rightarrow \text{3-diagonal matrix}\\ \text{fast, poor precision}}} = \underbrace{(\hat{H} - E^{(0)})\Psi^{(0)}}_{\substack{\text{Rayleigh-Ritz}\\ \text{good precision}}}$$

At each iteration: 3-diagonal linear system ⇒ $\underbrace{N \text{ operations}}_{\text{very fast}}$

Example: 100 iterations × 1% precision ⇒ $\underline{\delta\Psi \sim \delta E \sim 10^{-200}}$!

Implementing the method

linearized Schrödinger equation

$$\underbrace{(\hat{H} - E^{(0)})\delta\Psi - \delta E\Psi^{(0)}}_{\text{LHS}} = \underbrace{(\hat{H} - E^{(0)})\Psi^{(0)}}_{\text{RHS}}$$

- **Right-hand side:** pseudospectral method

  1. Finite Box: $\underbrace{\boxed{\Psi(0) = \Psi(L) = 0}}$

     Exp-small error $\sim \mathrm{e}^{-c\cdot L^{\alpha}}$ under barrier

  2. Use Fourier: $\boxed{\Psi(x) = \sum_{k=1}^{N} \underbrace{\tilde{\Psi}_{p_k}}_{\text{new vars}} \sin\Big(\underbrace{\frac{\pi k}{L}}_{p_k}\, x\Big)}$

     Exp-small error: $\tilde{\Psi}_{p_N} \sim \mathrm{e}^{-c' p_N}$ ← smooth $\Psi(x)$

     $\underbrace{\text{FFT:}}_{\text{superfast!}}$ $\{\tilde{\Psi}_{p_k}\} \longleftrightarrow \{\Psi(x_j)\}$ $\underset{\text{uniform lattice}}{\boxed{x_j = j\Delta x = jL/N}}$

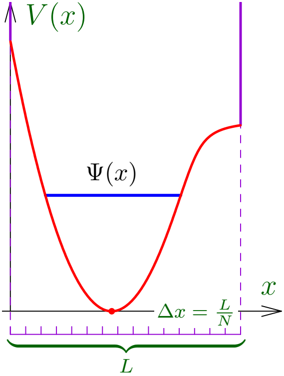
  3. Calculate RHS: $\left.\begin{array}{l} -\partial_x^2\Psi(x_j) = \widetilde{\mathrm{FFT}}_j\left[p_k^2\,\tilde{\Psi}_{p_k}\right] \\ V(x_j)\Psi(x_j) \text{ — multiply} \end{array}\right\} \Rightarrow$

     $(\hat{H} - E^{(0)})\Psi^{(0)}$

     $N \ln N$ operations

     Exp-small error!

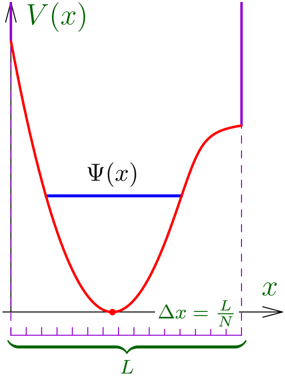
     Use multiprecision package!



Implementing the method

linearized Schrödinger equation

$$\underbrace{(\hat{H} - E^{(0)})\delta\Psi - \delta E\Psi^{(0)}}_{\text{LHS}} = \underbrace{(\hat{H} - E^{(0)})\Psi^{(0)}}_{\text{RHS}}$$



- **Left-hand side:**
  1. Usual (double-precision) arithmetics
  2. N-site lattice $\{x_j\}$, $\Delta x = L/N$ for $\delta\Psi$
  3. Second-order discretization:

     $$\partial_x^2\,\delta\Psi_j \to \frac{1}{\Delta x^2}\left(\delta\Psi_{j+1} + \delta\Psi_{j-1} - 2\delta\Psi_j\right)$$

     $\Rightarrow [\hat{H} - E^{(0)}]$ is a 3–diagonal matrix
  4. Solve 3–diagonal linear system

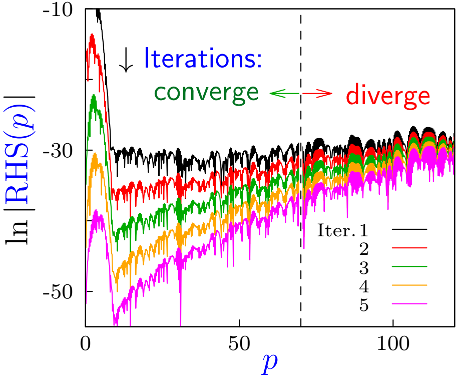
     $$\Rightarrow \boxed{\delta\Psi \text{ in } N \underbrace{\text{fast}}_{\text{usual (double-precision) numbers}} \text{operations!}}$$

- **Iterations:** $$\left.\begin{array}{l}\Psi^{(0)} \to \Psi^{(0)} + \delta\Psi \\ E^{(0)} \to E^{(0)} + \delta E\end{array}\right\} \Rightarrow$$

operations: $N \ln N \times \#_{\text{iterations}}$

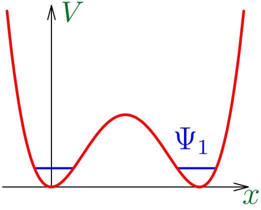
Fast: almost like lattice methods!

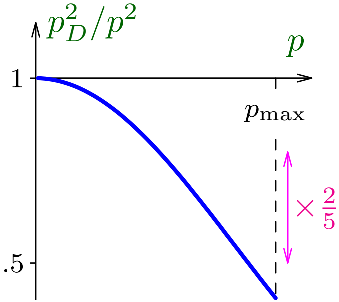
But this fails!



$$V(x) = \frac{1}{2}\,x^2(1-gx)^2$$

$g = 0.05$





Reason: lattice fails at high p!

$$-\partial_x^2\delta\Psi\Big|_{\text{lattice}} \xrightarrow{\text{Fourier}} p_D^2\,\delta\Psi_p \equiv \underbrace{p^2\mathrm{sinc}^2\left(\frac{p\Delta x}{2}\right)}_{\text{lattice momentum}^2}\delta\Psi_p$$

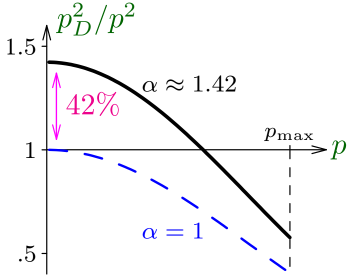
Simple cure

$$-\partial_x^2\delta\Psi_j\Big|_{\text{LHS}} = -\alpha\cdot\frac{\delta\Psi_{j+1}+\delta\Psi_{j-1}-2\delta\Psi_j}{\Delta x^2}$$

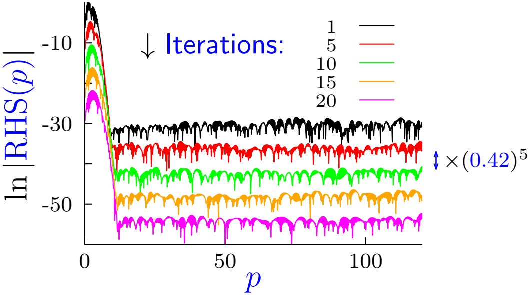
$$\Downarrow$$

$$p_D^2 = \alpha\cdot p^2\,\mathrm{sinc}^2\left(\frac{p\Delta x}{2}\right)$$

$$\boxed{\min_{\alpha}\left\|\frac{p_D^2-p^2}{p^2}\right\| \approx 0.42}$$ is at $\alpha = \frac{2\pi^2}{\pi^2+4}$



This is enough to reach slow convergence!



Receipt for convergence

Fit $-\partial_x^2$ with 3-diag matrix

at all p

Upgrade the method

Make $p_D \approx p$ at all momenta

- **Calculate LHS on 2N-lattice:**

$$\begin{array}{l} \Delta x \to \frac{1}{2}\Delta x \\ N \to 2N \end{array} \Rightarrow \left| \begin{array}{l} [+]\ p_D \text{ is closer to } p \text{ — faster convergence!} \\ [-]\ \times 2 \text{ operations — longer iterations!} \end{array} \right.$$

NB. Even larger lattice does not help (too long iterations)

- **Numerov method:** *Numerov '1924*

$$\partial_x^2 \delta\Psi_j\Big|_{\rm LHS} = \alpha\, \frac{\delta\Psi_{j+1}+\delta\Psi_{j-1}-2\delta\Psi_j}{\Delta x^2} + \beta\, (\underbrace{\delta\Psi''_{j+1} + \delta\Psi''_{j-1} - 2\delta\Psi''_j}_{\Delta x^2 \delta\Psi''''_j})$$

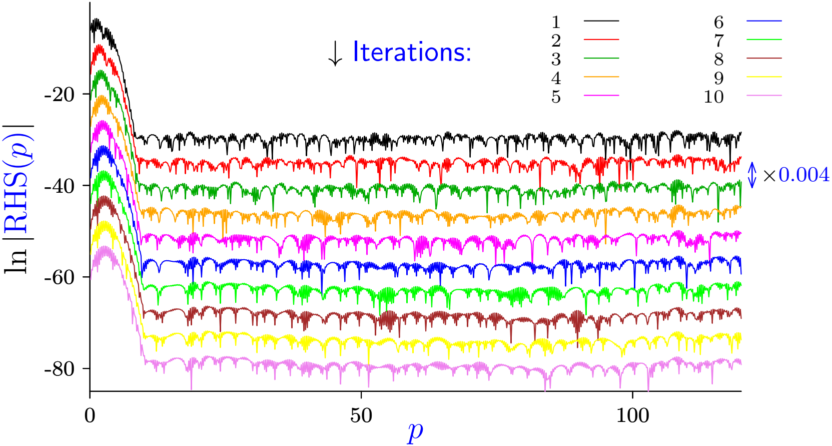
Three-diagonal: use Schrödinger Eq. $\delta\Psi''_{j+1} = 2(V_{j+1} - E^{(0)})\delta\Psi_{j+1} + \dots$

Original method: 4th order with $\alpha = 1$, $\beta = 1/12$

Our case: $$p_D^2 = \frac{4\,\alpha\, p^2 \mathrm{sinc}^2(\frac{p\Delta x}{4})}{4 - \beta \Delta x^2 p^2 \mathrm{sinc}^2(\frac{p\Delta x}{4})}$$ — 2 parameters to fit!

$$\min_{\alpha,\,\beta} \left\| \frac{p_D^2 - p^2}{p^2} \right\| \approx \boxed{0.004}$$ is at $\alpha \approx 0.996$, $\beta = \frac{1}{2} - \frac{4}{\pi^2}$

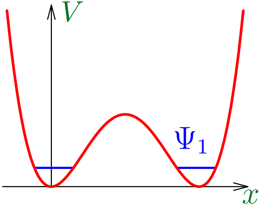
It works!



Each iteration RHS $\to 5\cdot 10^{-3}$ RHS

Computational price

$V(x) = \frac{1}{2}\, x^2 (1 - gx)^2$



- **In practice:** consider $\Psi_1(x)$ at $g = 0.05$

  Need $\boxed{\textbf{1.5 min}}$ $L = 80,\ N = 2048$

  for $\boxed{\textbf{1000 digits}}$ **of** Ψ_1

  Universal, no other algorithm is needed

- **In theory:** precision $\boxed{\epsilon = 10^{-D}}$

$$\Psi(L) \sim \underbrace{\mathrm{e}^{-gL^3/3}}_{\text{under}-\text{barrier}} < \delta \;\Rightarrow\; L \sim \left|\frac{3}{g}\ln\delta\right|^{1/3} \propto D^{1/3}$$

$$\Psi_{p_{\max}} \sim \underbrace{\mathrm{e}^{-\sqrt{2p_{\max}^3/9g}}}_{\text{saddle}-\text{point}} < \delta \;\Rightarrow\; p_{\max} \equiv \frac{\pi}{\Delta x} \sim \left|\frac{9g}{2}(\ln\delta)^2\right|^{1/3} \propto D^{2/3}$$

$$\Rightarrow\; N \sim \frac{3}{2^{1/3}\pi}\,|\ln\delta| \propto D$$

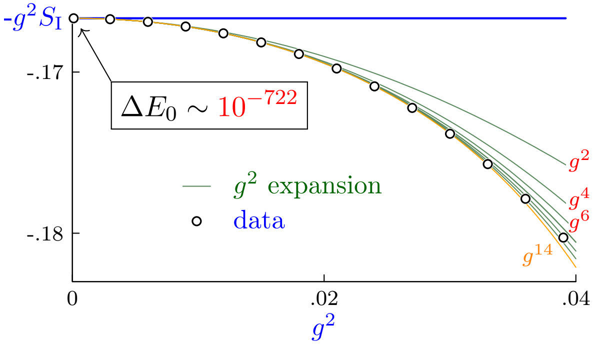
$$\boxed{(N\ln N\ \textbf{operations}) \times (\tfrac{D}{2}\ \textbf{iterations}) \propto D^2 \ln D}$$

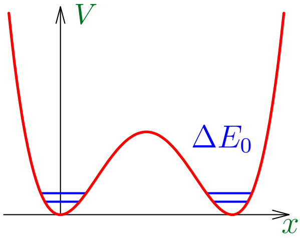
computational const

- **Other universal algorithms:** $N^2 \propto D^3$ in multiprecision

Application 1: Energy splitting

$V(x) = x^2(1-gx)^2/2$





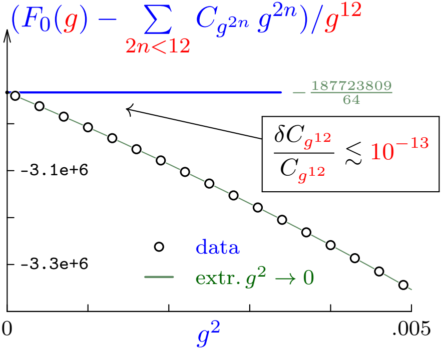
$\boxed{g^2 \ll 1}$ — semiclassical constant

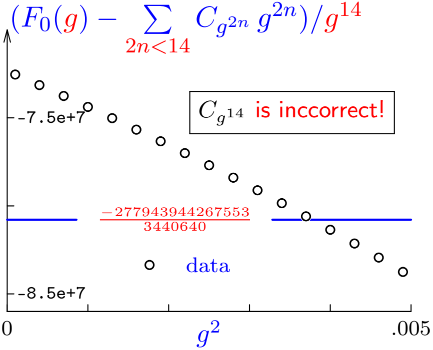
Zinn-Justin, Jentschura '2004

$$\Delta E_0 = \frac{2}{\sqrt{\pi g}} \exp\left\{ \overbrace{-\frac{1}{6g^2}}^{-S_I \leftarrow \text{instanton}} \underbrace{- \frac{71}{12} g^2 - \frac{315}{8} g^4 - \frac{65953}{144} g^6 - \frac{455525}{64} g^8 - \frac{64433873}{480} g^{10} \boxed{-\frac{187723809}{64}} g^{12} + \boxed{-\frac{277943944267553}{3440640}} g^{14} + O(g^{16})}_{\text{fluctuations}} \right\}$$

$$= \frac{2}{\sqrt{\pi g}}\, \mathrm{e}^{-\boxed{F_0(g)}/g^2}$$

Application 1: Energy splitting





Zinn-Justin, Jentschura '2004

$$\Delta E_0 = \frac{2}{\sqrt{\pi g}} \exp\left\{ -\frac{1}{6g^2} - \frac{71}{12} g^2 - \frac{315}{8} g^4 - \frac{65953}{144} g^6 - \frac{455525}{64} g^8 - \frac{64433873}{480} g^{10} \underbrace{\boxed{-\frac{187723809}{64}}\, g^{12} + \cancel{\boxed{-\frac{277943944267553}{3440640}}}\, g^{14} + O(g^{16})}_{\text{fluctuations}} \right\}$$

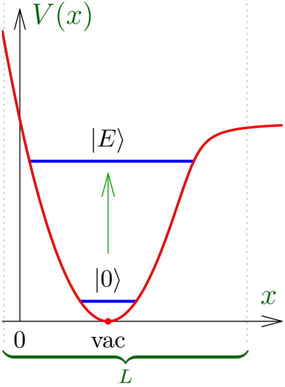
$$= \frac{2}{\sqrt{\pi g}}\, e^{-\boxed{F_0(g)}/g^2}$$

Also: breaks Dunne-Unsal relation! *Dunne, Ünsal '2013*

Application 2: Matrix elements

$$V = \frac{1}{2g^2}\left(\sqrt{(gx-1)^2+a^2} - gx\right)^2$$

$a = 0.1$ $\boxed{g^2 \ll 1}$ — semicl. constant



$$\boxed{\langle E|\hat{x}|0\rangle = \int dx \, \underbrace{x\Psi_E(x)\Psi_0(x)}_{\text{oscillating function!}}}$$

Semiclassics (Landau method): cancellations!

See talk by R. Kolosov's

$$\langle E|\hat{x}|0\rangle = g^{7/2}\,\mathrm{Re}\,\underbrace{\mathrm{e}^{-\mathcal{W}_s/g^2}}_{\text{exp. small}}\left[\mathcal{A} + \underbrace{\mathcal{O}(g^2)}_{g\to 0!}\right]$$

Provides expressions for $\mathcal{W}_s(E)$, $\mathcal{A}(E)$.

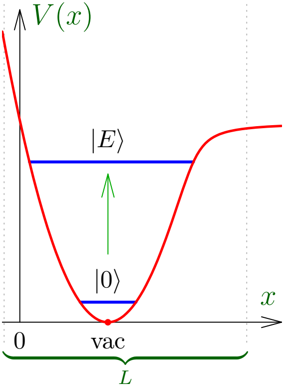
Test the theory: $\langle E|\hat{x}|0\rangle = \sum_{n_p=1}^{N} \underbrace{[\Psi_E]_p\,[x\Psi_0]_{-p}}_{\text{known precisely!}}$

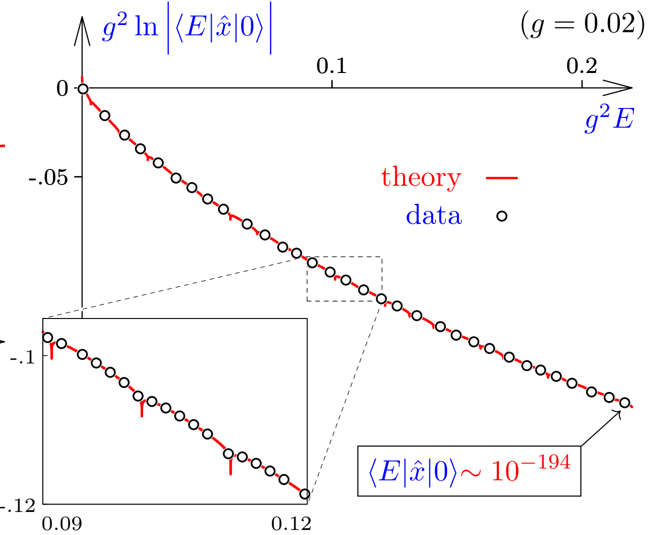
$\Rightarrow$ **Check the theory!**

Application 2: Matrix elements

$$V = \frac{1}{2g^2}\left(\sqrt{(gx-1)^2+a^2} - gx\right)^2$$

$a = 0.1$ $\boxed{g^2 \ll 1}$ — semicl. constant





$\langle E|\hat{x}|0\rangle \sim 10^{-194}$

Conclusions & further applications

Our method is:

- Precise: 10^2-10^3 digits ← exp-small precision
- Generic: works for any smooth $V(x)$
- Fast: $\underbrace{N \ln N}_{\text{FFT}} \times \underbrace{(N_{\text{digits}}/2)}_{\text{iterations}}$ | [−] extra iterations
  | [+] small N is enough for good precision

  → Faster than other generic methods

  → Slower than recurrent methods for polynomial $V(x)$
- Simple: special lattice discretization + FFT + iterations

  → Can be used in many lattice codes

Further applications:

- Schrödinger equation in 2- and 3 dimensions with multiprecision → Test of exact WKB & resurgence
- Boundary value problems: ordinary precision but smaller N!
- Nonlinear equations: multiprecision & 1, 2, and 3 dimensions

Thank you for attention!

*The talk is supported by the grant RSCF 22-12-00215-П