Efficient Calculation of the Two-Dimensional Wigner Potential

P. Ellinghaus, M. Nedjalkov, and S. Selberherr
Institute for Microelectronics, TU Wien, Gußhausstraße 27–29/E360, 1040 Wien, Austria
E-mail: {ellinghaus | nedjalkov | selberherr}@iue.tuwien.ac.at

The calculation of the Wigner potential (WP) in two-dimensional simulations consumes a considerable part of the computation time. A reduction of the latter is therefore very desirable, in particular for self-consistent solutions, where the WP must be recalculated many times. We introduce an algorithm – named box discrete Fourier transform (BDFT) – that reduces the computational effort roughly by a factor of five.

The semi-discrete Wigner potential is defined as

$$V_W\left(\mathbf{r}, \mathbf{P\Delta k}\right) \equiv \frac{1}{i\hbar\mathbf{L}} \int_{-\mathbf{L}/2}^{\mathbf{L}/2} d\mathbf{s}\, e^{-i2\mathbf{P\Delta k}\cdot\mathbf{s}} \delta V \quad (1)$$

$$\delta V\left(\mathbf{s}; \mathbf{r}\right) \equiv V\left(\mathbf{r}+\mathbf{s}\right) - V\left(\mathbf{r}-\mathbf{s}\right), \quad (2)$$

where $\mathbf{s}$ is bounded by a finite coherence length, $\mathbf{L}$. The momentum vector $\mathbf{P\Delta k}$ is discretized in steps of $\Delta k = \frac{\pi}{\mathbf{L}}$. The discretization of position vectors $\mathbf{r}$ and $\mathbf{s}$ yields our 2D computational domain (Fig. 1) for which the fully discretized WP,

$$V_W\left(x, y, p, q\right) = \frac{1}{i\hbar MN} \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} e^{-i2\Delta k(pm\Delta x + qn\Delta y)} \delta V\left(x \pm \left(m - \frac{M}{2}\right)\Delta x, y \pm \left(n - \frac{N}{2}\right)\Delta x\right), \quad (3)$$

must be calculated at each mesh node.

Eq. (3) is akin to a DFT of (2), conventionally calculated using an FFT algorithm, which has $\mathcal{O}(N \log_2 N)$ complexity. To derive our BDFT algorithm, we adopt the idea of the one-dimensional sliding DFT [2], [3], which has $\mathcal{O}(N)$ complexity. The algorithm calculates the Fourier coefficients of a sequence $\{x_c..x_{c+N-1}\}$ using the coefficients calculated for $\{x_{c-1}..x_{c+N-2}\}$:

$$X_c(p) = e^{i\frac{2\pi p}{N}} \left(X_{c-1}(p) + x_{c+N-1} - x_{c-1}\right). \quad (4)$$

The two sequences differ by a single value, just like each row/column of the coherence boxes of adjacent nodes. We exploit this observation and calculate the DFT of the first N values of all M' rows in the domain, using an FFT algorithm. In the resulting $M' \times N$ matrix of Fourier coefficients, we calculate the DFT of the first M values of every column. After this initialization, $V_w\left(0, 0, p, q\right)$ is known and we start moving the coherence box in a column-wise manner across the domain and apply (4) to calculate the row/column DFTs as needed. This initialization approach shows favourable serial performance and cache complexity; however, a parallelized implementation would require multiple (modified) initializations.

To allow the application of (4) to calculate (3), we have to reformulate (1) using a substitution of variables (Fourier shift theorem), such that

$$V_W\left(\mathbf{r}, \mathbf{P\Delta k}\right) = \frac{2}{\hbar\mathbf{L}} \mathbb{I}\mathrm{m} \int_{-\mathbf{L}/2}^{\mathbf{L}/2} d\mathbf{s} e^{-i2\mathbf{P\Delta k}\cdot\mathbf{s}} V\left(\mathbf{r}+\mathbf{s}\right)$$

This formulation has the further advantage of halving the computation time by avoiding to calculate the DFT of the potential difference. We also note that (4) allows, unlike the FFT, to easily compute only selected momentum (p) values (under justified physical assumptions), thereby offering a further possibility to reduce the computational costs.

Fig. 2 shows a potential profile for which the WP (Fig. 3) has been calculated and demonstrates the expected anti-symmetry. Fig. 4 shows the WP for fixed values of x and $p\Delta k_x$. The BDFT algorithm was benchmarked against an FFT implementation using the FFTW library [1], with a setup detailed in Table II. Table I makes evident that the BDFT reduces the computation time by at least a factor of five. The performance of the FFT algorithms selected by the FFTW library strongly depends on the transform size (coherence length), whereas the BDFT algorithm scales at a constant rate with size.

ISBN 978-2-9547858-0-6

ACKNOWLEDGEMENT

This work has been supported by the Austrian Science Fund, project FWF-P21685-N22.

REFERENCES

[1] M. Frigo and S.G. Johnson, *The Design and Implementation of FFTW3*, Proceedings of the IEEE, **93**, 216 (2005).

[2] E. Jacobsen and R. Lyons, *The sliding DFT*, Signal Processing Magazine, IEEE, **20**, 74 (2003).

[3] E. Jacobsen and R. Lyons, *An update to the sliding DFT*, Signal Processing Magazine, IEEE, **21**, 110 (2004).

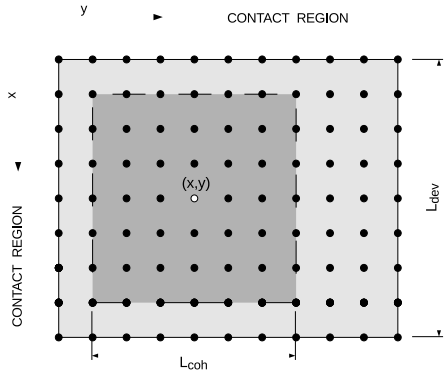

Fig. 1. Coherence box of size $\mathbf{L}_{coh} = (M\Delta x, N\Delta y)$, centred at (x, y) in the discretized domain, of size $\mathbf{L}_{dev} = (M'\Delta x, N'\Delta y)$, surrounded by semi-infinite contact regions. .

TABLE I

COMPUTATION TIME OF 2D WIGNER POTENTIAL

L_{dev} [a.u.]	L_{coh} [a.u.]	BDFT [s]	FFT [s]	Speed-up [1]
100	100	0.12	0.75	6.3
200	100	0.47	2.53	5.4
300	99	0.96	10.66	11.1
300	100	1.00	5.77	5.8
300	101	1.04	61.25	55.9
400	100	1.78	10.27	5.8
500	100	2.80	17.84	6.4

TABLE II

BENCHMARK AND SETUP SPECIFICS

Hardware	Intel Core 3110M; 8 GB (dual channel)
OS	Ubuntu 13.10 (64 bit)
Compiler	gcc 4.8.1
flags	-O3 -fastmath -march=native
FFT library	FFTW 3.3 (SIMD enabled)
interface \ flags	dft_r2c_2d \ FFTW_MEASURE

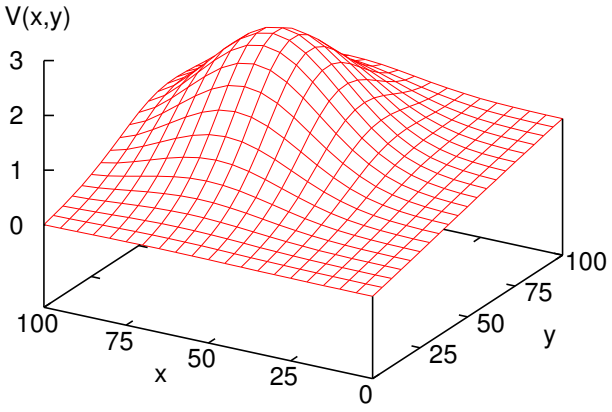

Fig. 2. A Gaussian potential (allows an analytic expression, for comparison) is superimposed on a ramp and represents the profile used in the presented calculations.

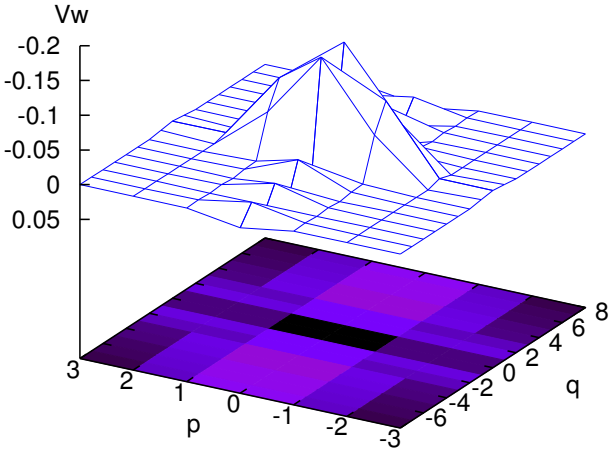

Fig. 3. The normalized Wigner potential, $V_w\,(x = 50, y = 50, p, q)$ with the expected anti-symmetry, $V_w(\cdot, p, q) = V_w(\cdot, -p, -q)$, as seen in the colour map.

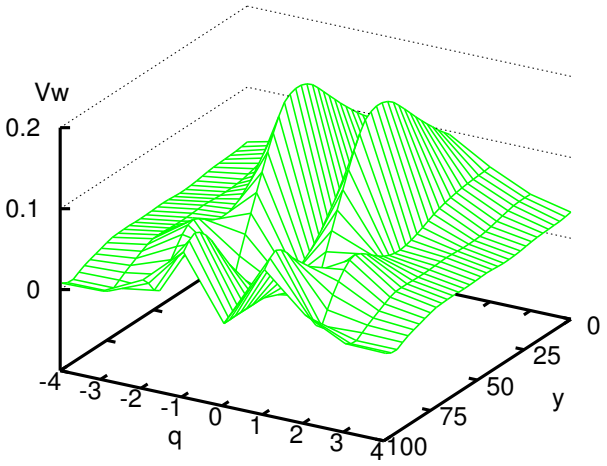

Fig. 4. The normalized Wigner potential, $V_w\,(x = 50, y, p = 0, q)$, for fixed values of x and $p\Delta k_x$.