ETH
Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

Seminar for Applied Mathematics SAM

Numerical Steepest Descent for Overlap Integrals of Semiclassical Wavepackets

R. Bourquin and V. Gradinaru

Research Report No. 2015-12
April 2015

Seminar für Angewandte Mathematik
Eidgenössische Technische Hochschule
CH-8092 Zürich
Switzerland

Funding SNF: 140688

Numerical Steepest Descent for Overlap Integrals of Semiclassical Wavepackets

R. Bourquin and V. Gradinaru

April 30, 2015

Abstract

In this report we show that classical Gauss quadrature is not adequate for a large class of correlation and overlap integrals originating from quantum mechanics. These integrals are usually highly oscillatory and therefore special methods are necessary for accurate computations. We develop and test a new, highly efficient method based on some recent results about numerical steepest descent to solve the problem stated. Our approach is built in principle to work for any number of space dimensions but some care must be taken not to run into the curse of dimensionality. Explicit formulae and algorithms are given in full generality.

1 Introduction

Consider smooth functions $f(\underline{x})$ and $g(\underline{x})$ with $\underline{x} \in \mathbb{R}^N$, we compute the integral:

$$I := \int_{-\infty}^{\infty} \cdots \int_{-\infty}^{\infty} f(\underline{x}) \, \exp(i\omega g(\underline{x})) \, \mathrm{d}\underline{x} \tag{1}$$

where $\omega > 0$ is the *frequency*. This is an instance from a very large and important class of highly oscillatory integrals. Usually one calls the function g the *oscillator* and the function f the *amplitude* or *envelope*.
Computing the integral above by direct standard numerical quadrature schemes would look like depicted in Figure 1 where we apply a suitable quadrature rule, in our specific case Gauss-Hermite rules, directly to the integral I. Since our interests stem from quantum mechanics, we anticipate the integrand in I being composed of wavepackets ϕ and ϕ', forming overlap integrals like $\langle\phi|\phi'\rangle$.

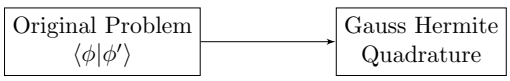


Figure 1: Overview of the integral computation.

Direct quadrature in general requires that the number of quadrature points increases with ω, making it more and more expensive the larger the oscillation frequency gets. In our case this growth continues up to a range that we can not afford. It turns out that we need much better methods to efficiently deal with

this kind of integrals and obtain mathematically correct results. The theory of *numerical steepest descent* was developed about ten years ago exactly for these integrals. For more details on the mathematical theory see the work by Huybrechs, Vandewalle and others. The main technique is described in [15] for the one-dimensional case and in [16] for multivariate integrands. Semi-finite intervals appear in [17]. Other bits and pieces can be found in [7] and [2].
In this text we will concentrate on the application of this theory on the computation of some overlap integrals from quantum mechanics.

2 About semiclassical wavepackets

Semiclassical wavepackets in D dimensions are constructed from a groundstate:

$$\phi_{\underline{0}}(\underline{x}) = (\pi\varepsilon^2)^{-\frac{D}{4}} (\det \mathbf{Q})^{-\frac{1}{2}} \exp\left(\frac{\imath}{2\varepsilon^2} \left\langle (\underline{x}-\underline{q}), \mathbf{P}\mathbf{Q}^{-1}(\underline{x}-\underline{q})\right\rangle + \frac{\imath}{\varepsilon^2}\left\langle \underline{p}, (\underline{x}-\underline{q})\right\rangle\right)$$

by raising and lowering operators $\mathcal{R}$ and $\mathcal{L}$ and enumerated by a multi-index vector $\underline{k} \in \mathbb{N}_0^D$. There are the parameters for average position $\underline{q} \in \mathbb{R}^D$ and momentum $\underline{p} \in \mathbb{R}^D$. Additionally there are two invertible complex matrices $\mathbf{Q} \in \mathbb{C}^{D\times D}$ and $\mathbf{P} \in \mathbb{C}^{D\times D}$ satisfying the Conditions:

$$\mathbf{Q}^{\mathrm{T}}\mathbf{P} - \mathbf{P}^{\mathrm{T}}\mathbf{Q} = \mathbf{0} \tag{2}$$

$$\mathbf{Q}^{\mathrm{H}}\mathbf{P} - \mathbf{P}^{\mathrm{H}}\mathbf{Q} = 2\imath\mathbf{I}\,. \tag{3}$$

Implied by these constraints, the matrix $\mathbf{P}\mathbf{Q}^{-1}$ is complex symmetric (but not Hermitian) and has positive definite imaginary part. We collect all these parameters in the set $\Pi := \{\underline{q}, \underline{p}, \mathbf{Q}, \mathbf{P}\}$. Further there is the semiclassical scaling parameter $1 \gg \varepsilon > 0$. For many more details describing these wavepackets, see [12] and [4]. What matters most here is, that each wavepacket ϕ is of the form:

$$\phi[\Pi](\underline{x}) \sim p(\underline{x}) \exp\left(\frac{\imath}{\varepsilon^2} g(\underline{x})\right) \tag{4}$$

with $p(\underline{x})$ a multivariate polynomial. This form is sufficient for the steepest descent technique to be applicable. Indeed, comparing to the integrand in (1), we find that $\omega = \frac{1}{\varepsilon^2}$ which means that the oscillator frequency increases very fast for small scaling parameters ε. The steepest descent method which we will look at in the next section becomes the *better* the smaller ε is. The actual oscillator g is of the form:

$$g(\underline{x}) := \frac{1}{2}\left\langle \underline{x}-\underline{q}, \mathbf{P}\mathbf{Q}^{-1}\left(\underline{x}-\underline{q}\right)\right\rangle + \left\langle \underline{p}, \underline{x}-\underline{q}\right\rangle \tag{5}$$

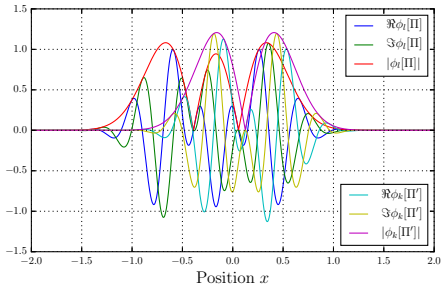
We are interested in overlap integrals[1] like $\langle \phi_k, \phi_l\rangle$. These integrals become even more difficult in case the parameters sets Π_k and Π_l differ. In any case the

[1] For one-dimensional wavepackets there exists an exact formula which is very expensive to evaluate numerically, see the original work [13] and [3] for compatible notation.

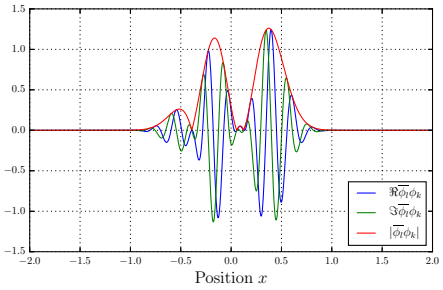
integral is still of the same oscillatory structure like a single wavepacket:

$$\begin{aligned}\langle \phi_k, \phi_l \rangle &= \int_{-\infty}^{\infty} \overline{p_k(x) \exp\left(\frac{\imath}{\varepsilon^2} g_k(x)\right)} p_l(x) \exp\left(\frac{\imath}{\varepsilon^2} g_l(x)\right) \mathrm{d}\underline{x} \\ &= \int_{-\infty}^{\infty} \overline{p_k(x)} p_l(x) \exp\left(-\frac{\imath}{\varepsilon^2} \overline{g_k(x)} + \frac{\imath}{\varepsilon^2} g_l(x)\right) \mathrm{d}\underline{x} \\ &= \int_{-\infty}^{\infty} \overline{p_k(x)} p_l(x) \exp\left(\frac{\imath}{\varepsilon^2} \left(-\overline{g_k(x)} + g_l(x)\right)\right) \mathrm{d}\underline{x} \,.\end{aligned} \tag{6}$$

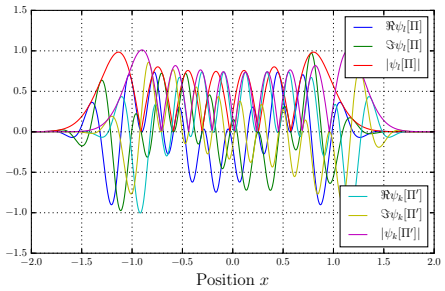
The next step to take is to combine both oscillators g_k and g_l into a single one such that we get back a formal expression like the one in (1) we started with. But first we would like to show in Figure 2 a concrete example of how the wavepackets involved and the integrand look like.



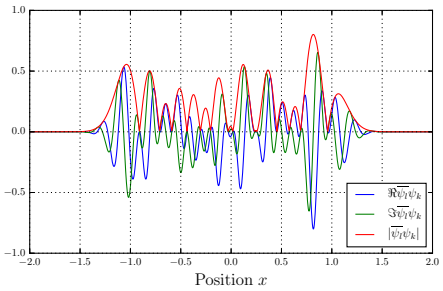
(a) Wavepackets ϕ_1 and ϕ_2.



(b) Integrand in case of ϕ_1 and ϕ_2.



(c) Wavepackets ϕ_8 and ϕ_6.



(d) Integrand in case of ϕ_8 and ϕ_6.

Figure 2: Examples of wavepackets ϕ_k and ϕ_l and the integrand $\overline{\phi_k}\phi_l$. The parameter sets are $q_k = \frac{1}{8}$, $p_k = -\frac{1}{2}$, $Q_k = \frac{9}{10}$, $P_k = \frac{10}{9}\imath$ and $q_l = -\frac{1}{6}$, $p_l = \frac{2}{5}$, $Q_l = 1$, $P_l = \imath$. The scaling parameter is $\varepsilon = \frac{1}{\sqrt{10}} \approx 0.316$. Note that the indices k and l typically range up to a few tens and this ε is not very small.

2.1 Combining different oscillators

Combining the two oscillators is actually a straight-forward computation, however it is very error prone, big chances are that we miss a transpose or conjugate somewhere. To simplify the notation we define the matrix:

$$\mathbf{\Gamma_i} := \mathbf{P_i} \mathbf{Q_i}^{-1} \tag{7}$$

and start with computing $\overline{g_k}$:

$$\begin{aligned}
\overline{g_k} &= \frac{1}{2}\overline{\langle \underline{x} - \underline{q_k}, \mathbf{\Gamma_k}\left(\underline{x} - \underline{q_k}\right)\rangle} + \overline{\langle \underline{p_k}, \underline{x} - \underline{q_k}\rangle} \\
&= \frac{1}{2}\left(\overline{\langle \underline{x}, \mathbf{\Gamma_k}\underline{x}\rangle} - \overline{\langle \underline{x}, \mathbf{\Gamma_k}\underline{q_k}\rangle} - \overline{\langle \underline{q_k}, \mathbf{\Gamma_k}\underline{x}\rangle} + \overline{\langle \underline{q_k}, \mathbf{\Gamma_k}\underline{q_k}\rangle}\right) + \overline{\langle \underline{p_k}, \underline{x}\rangle} - \overline{\langle \underline{p_k}, \underline{q_k}\rangle} \\
&= \frac{1}{2}\left(\underline{x}^{\mathrm{H}}\mathbf{\Gamma_k}^{\mathrm{H}}\underline{x} - \underline{q_k}^{\mathrm{H}}\mathbf{\Gamma_k}^{\mathrm{H}}\underline{x} - \underline{x}^{\mathrm{H}}\mathbf{\Gamma_k}^{\mathrm{H}}\underline{q_k} + \underline{q_k}^{\mathrm{H}}\mathbf{\Gamma_k}^{\mathrm{H}}\underline{q_k}\right) + \underline{x}^{\mathrm{H}}\underline{p_k} - \underline{q_k}^{\mathrm{H}}\underline{p_k}\,.
\end{aligned}$$

Next we add g_l and simplify the following expression:

$$\begin{aligned}
-\overline{g_k} + g_l =& -\frac{1}{2}\left(\underline{x}^{\mathrm{H}}\mathbf{\Gamma_k}^{\mathrm{H}}\underline{x} - \underline{q_k}^{\mathrm{H}}\mathbf{\Gamma_k}^{\mathrm{H}}\underline{x} - \underline{x}^{\mathrm{H}}\mathbf{\Gamma_k}^{\mathrm{H}}\underline{q_k} + \underline{q_k}^{\mathrm{H}}\mathbf{\Gamma_k}^{\mathrm{H}}\underline{q_k}\right) - \left(\underline{x}^{\mathrm{H}}\underline{p_k} - \underline{q_k}^{\mathrm{H}}\underline{p_k}\right) \\
&+ \frac{1}{2}\left(\underline{x}^{\mathrm{H}}\mathbf{\Gamma_l}\underline{x} - \underline{x}^{\mathrm{H}}\mathbf{\Gamma_l}\underline{q_l} - \underline{q_l}^{\mathrm{H}}\mathbf{\Gamma_l}\underline{x} + \underline{q_l}^{\mathrm{H}}\mathbf{\Gamma_l}\underline{q_l}\right) + \left(\underline{p_l}^{\mathrm{H}}\underline{x} - \underline{p_l}^{\mathrm{H}}\underline{q_l}\right) \\
=& \frac{1}{2}\left(\underline{x}^{\mathrm{H}}\mathbf{\Gamma_l}\underline{x} - \underline{x}^{\mathrm{H}}\mathbf{\Gamma_k}^{\mathrm{H}}\underline{x}\right) + \frac{1}{2}\left(\underline{q_k}^{\mathrm{H}}\mathbf{\Gamma_k}^{\mathrm{H}}\underline{x} + \underline{x}^{\mathrm{H}}\mathbf{\Gamma_k}^{\mathrm{H}}\underline{q_k} - \underline{x}^{\mathrm{H}}\mathbf{\Gamma_l}\underline{q_l} - \underline{q_l}^{\mathrm{H}}\mathbf{\Gamma_l}\underline{x}\right) \\
&+ \left(\underline{p_l}^{\mathrm{H}}\underline{x} - \underline{x}^{\mathrm{H}}\underline{p_k}\right) + \frac{1}{2}\left(\underline{q_l}^{\mathrm{H}}\mathbf{\Gamma_l}\underline{q_l} - \underline{q_k}^{\mathrm{H}}\mathbf{\Gamma_k}^{\mathrm{H}}\underline{q_k}\right) + \left(\underline{q_k}^{\mathrm{H}}\underline{p_k} - \underline{p_l}^{\mathrm{H}}\underline{q_l}\right) \\
=& \frac{1}{2}\underline{x}^{\mathrm{H}}\left(\mathbf{\Gamma_l} - \mathbf{\Gamma_k}^{\mathrm{H}}\right)\underline{x} + \frac{1}{2}\left(\underline{q_k}^{\mathrm{H}}\mathbf{\Gamma_k}^{\mathrm{H}}\underline{x} - \underline{q_l}^{\mathrm{H}}\mathbf{\Gamma_l}\underline{x} + \underline{q_k}^{\mathrm{T}}\overline{\mathbf{\Gamma_k}}\overline{\underline{x}} - \underline{q_l}^{\mathrm{T}}\mathbf{\Gamma_l}^{\mathrm{T}}\overline{\underline{x}}\right) \\
&+ \left(\underline{p_l}^{\mathrm{H}}\underline{x} - \underline{p_k}^{\mathrm{T}}\overline{\underline{x}}\right) + \frac{1}{2}\left(\underline{q_l}^{\mathrm{H}}\mathbf{\Gamma_l}\underline{q_l} - \underline{q_k}^{\mathrm{H}}\mathbf{\Gamma_k}^{\mathrm{H}}\underline{q_k}\right) + \left(\underline{q_k}^{\mathrm{H}}\underline{p_k} - \underline{p_l}^{\mathrm{H}}\underline{q_l}\right)\,.
\end{aligned}$$

In the end we find a general quadratic oscillator of the common expanded normal form $g(\underline{x}) = \underline{x}^{\mathrm{H}}\mathbf{A}\underline{x} + \underline{b}^{\mathrm{T}}\underline{x} + c$ where careful computation gives:

$$\begin{aligned}
\mathbf{A} &= \frac{1}{2}\left(\mathbf{\Gamma_l} - \mathbf{\Gamma_k}^{\mathrm{H}}\right) \\
\underline{b} &= \left(\frac{1}{2}\left(\underline{q_k}^{\mathrm{H}}\mathbf{\Gamma_k}^{\mathrm{H}} - \underline{q_l}^{\mathrm{H}}\mathbf{\Gamma_l} + \underline{q_k}^{\mathrm{T}}\overline{\mathbf{\Gamma_k}} - \underline{q_l}^{\mathrm{T}}\mathbf{\Gamma_l}^{\mathrm{T}}\right) + \left(\underline{p_l}^{\mathrm{H}} - \underline{p_k}^{\mathrm{T}}\right)\right)^{\mathrm{T}} \\
&= \frac{1}{2}\left(\overline{\mathbf{\Gamma_k}}\overline{\underline{q_k}} - \mathbf{\Gamma_l}^{\mathrm{T}}\overline{\underline{q_l}} + \mathbf{\Gamma_k}^{\mathrm{H}}\underline{q_k} - \mathbf{\Gamma_l}\underline{q_l}\right) + \left(\overline{\underline{p_l}} - \underline{p_k}\right) \\
c &= \frac{1}{2}\left(\underline{q_l}^{\mathrm{H}}\mathbf{\Gamma_l}\underline{q_l} - \underline{q_k}^{\mathrm{H}}\mathbf{\Gamma_k}^{\mathrm{H}}\underline{q_k}\right) + \left(\underline{q_k}^{\mathrm{H}}\underline{p_k} - \underline{p_l}^{\mathrm{H}}\underline{q_l}\right)
\end{aligned} \tag{8}$$

and we assumed that $\underline{x}$ is real. Be sure to pay attention to the fact that here we have $\underline{b}^{\mathrm{T}}$ only instead of $\underline{b}^{\mathrm{H}}$. The matrix $\mathbf{A}$ has a special property, both its real and imaginary parts are symmetric while the matrix itself is in general *not* Hermitian. This places the matrix right outside the convenient set of normal matrices, which implies that it is not diagonalizable by unitary matrices, an important consequence as we will see later.

3 Numerical steepest descent

3.1 Overview and summary of the ideas

A typical example of a highly oscillatory integral looks like:

$$I = \int_{\Omega} f(\underline{x})\,\exp(\imath\omega g(\underline{x}))\mathrm{d}\underline{x} \tag{9}$$

where the non-oscillatory function $g : \mathbb{R}^N \to \mathbb{C}$ is called the oscillator and the also non-oscillatory function $f : \mathbb{R}^N \to \mathbb{C}$ the envelope. The parameter $\omega \in \mathbb{R}^+$ is the frequency. Often one looks at the asymptotic behavior for $\omega \to \infty$. In our setting this parameter ω has a fixed and finite value. Finally there is the domain of integration denoted by $\Omega \subset \mathbb{R}^N$. In the theory shown so far this is a bounded subset of $\mathbb{R}^N$. Figure 3 shows a particular example of such a nasty integrand.

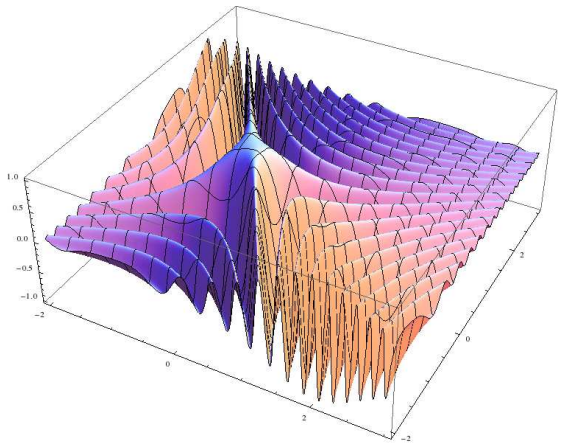


Figure 3: Oscillatory integrand of the integral $\int_{-2}^{3}\int_{-2}^{3} \frac{e^{5\imath(x^2-xy-y^2)}}{1+(x+y)^2}\mathrm{d}x\mathrm{d}y$ with envelope $f(x,y) = \frac{1}{1+(x+y)^2}$ and a quadratic oscillator $g(x,y) = x^2 - xy - y^2$ at low frequency of $\omega = 5$.

For our problem at hand, computing overlap integrals, we will need to integrate over the whole space in N dimensions and hence set $\Omega = \mathbb{R}^N$ in (1). There are two essential observations to be made about the oscillatory part $\exp(i\omega g(\underline{x}))$ of any such integral. First, this expression does decay exponentially fast for increasing $\Im g(z)$ and second, it does not oscillate for constant $\Re g(z)$. This can easily be seen by expanding complex numbers:

$$e^{\imath\omega g(z)} = e^{\imath\omega(\Re g(z)+\imath\Im g(z))} = e^{\imath\omega\Re g(z)}e^{-\omega\Im g(z)}\,.$$

The main idea behind the *numerical steepest descent* method is therefore to transform the integrand such that it is no longer oscillatory but rather exponentially decaying. For this we need to find a coordinate transformation $z = h(\tau)$ such that the *real* part of $g(z)$ is constant. In a second step we then apply Cauchy's Theorem for contour integrals along the path $h(\tau)$.
Let us look again at the plot in Figure 3 and ask the question in which regions of Ω the integrand contributes most to the value of the integral. By intuition one would say, at least asymptotically for $\omega \to \infty$, that oscillations in the integrand generally approximately cancel out and mainly places with locally no oscillations contribute to the value. Regions showing no oscillations are *stationary points* where $\nabla g(\underline{x}) = \underline{0}$ on one hand and so called *resonance points* defined by the condition $\nabla g(\underline{x}) \perp \partial\Omega$ on the other. In the one-dimensional case one can forget about the complications around resonance points and just take care of the endpoints of the interval $[a,b] = \Omega$. In our case there is no surface $\partial\Omega$ anyway as we compute over the whole space $\mathbb{R}^N$.

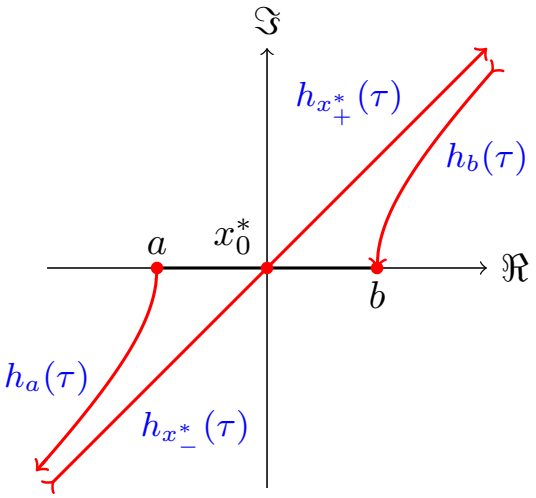


Figure 4: Example of an integral treated by the numerical steepest descent method. The oscillator is $g(x) = x^2$ and hence $\partial_x g(x) = 0$ at $x^* = 0$. Shown are the contributing points a, b and x_0^* and the paths $h_a(\tau)$, $h_b(\tau)$ and $h_{x_0^*,\pm}(\tau)$ attached to them. Instead of integrating directly from a to b along the real axis, we follow the red path through the complex plane.

In the following we briefly consider the one-dimensional integral over the interval $[a, b]$ and walk through the major steps involved in the procedure. We can find all stationary points $\{x_j^*\}_j$ from equating the gradient to zero (assuming the equation is indeed solvable):

$$\nabla_x \, g(x) = 0 \, . \tag{10}$$

The set of contributing points is then:

$$\Theta := \{a, b\} \cup \{x_j^*\}_j \, . \tag{11}$$

Next, we set up the so called *path equation* to compute the coordinate transformation $h(\tau)$ or equivalently the path of integration. This has to be done locally at all contributing points. Hence, for all $\xi \in \Theta$:

$$g(h_\xi(\tau)) = g(\xi) + \imath\tau \, . \tag{12}$$

Solving these equations yields a bunch of paths $h_\xi(\tau)$. Each one is indexed by the point ξ it starts at[2] and parametrised by $\tau \in \mathbb{R}_0^+$. Refer to Figure 4 for a simple example having only a single stationary point.
In general, solving the path equation amounts to finding an inverse of g. In some cases this inverse is multi-valued and the path not unique. At the endpoints we can choose the correct path $h_a(\tau)$ by requiring $h_a(0) = a$ which forces the path chosen to be really attached to the point a. At the stationary points we have to choose two paths, $h_{x_j^*,+}$ and $h_{x_j^*,-}$, one of which is considered incoming and the

[2]Note that this is not the full truth. In case of multi-valued inverses of g at the point ξ it can happen that $h_\xi(0) \neq \xi$. With $g(x) = x^2$ and $\xi = 3$ we find $h_3(\tau) = \pm\sqrt{9 + \imath\tau}$ and $h_3(0) = \pm 3 \neq 3$.

other outgoing (note that this is not reflected by the parametrisation in τ), such that the integration contour passes through each stationary point once. Then we have to make sure that the outgoing path of one stationary point and the incoming of the next one will lead to the same *valley*. This guarantees that we can concatenate all paths in the end and obtain a closed contour for integration. We will not bother too much with these details as they are not relevant for our problem. Many more details can be found in the original papers [15] and [16] by Huybrechs and Vandewalle. If the path equation can not be solved analytically, not all is lost as it is possible to work with numerical path approximations, see [1]. By the formal structure of the semiclassical wavepackets, the path equation in our case is always a multivariate quadratic form which we can solve explicitly. Now it is time to assemble the parts. For each path attached at a point $\xi \in \Theta$ we perform the transformation $x \mapsto h_\xi(\tau)$. Doing this change of variable will give a bunch of new integrals, one or two for each ξ:

$$\begin{aligned} J[\xi] &:= e^{\imath\omega g(\xi)} \int_0^\infty f(h_\xi(\tau))\, h'_\xi(\tau)\, e^{-\omega\tau}\, \mathrm{d}\tau \\ &= \frac{e^{\imath\omega g(\xi)}}{\omega} \int_0^\infty f\left(h_\xi\left(\frac{\tau}{\omega}\right)\right)\, h'_\xi\left(\frac{\tau}{\omega}\right)\, e^{-\tau}\, \mathrm{d}\tau\,. \end{aligned} \tag{13}$$

We continue by applying Cauchy's theorem:

$$I = e^{\imath\omega g(a)} J[a] + \sum_j \left(J[x^*_{j,+}] - J[x^*_{j,-}]\right) - e^{\imath\omega g(b)} J[b] \tag{14}$$

which is valid for the correct choice of paths. We glue together all the paths to a single long path connecting the endpoints a and b while wandering around in the complex plane and visiting each stationary point once. Obviously we have to be very careful, avoid crossing branch cuts and be aware of all potential poles. In our case there is no danger around because f is simply a polynomial.
Up to now we just transformed the problem, nothing more. The new task is to compute by quadrature all the integrals (13) and there can be exponentially many of these in higher dimensions. For this type of semi-finite integrals Gauss-Laguerre quadrature with nodes $\{\gamma_k\}_k$ and weights $\{w_k\}_k$ seems to be most appropriate:

$$J[\xi] \approx \frac{e^{\imath\omega g(\xi)}}{\omega} \sum_k f\left(h_\xi\left(\frac{\gamma_k}{\omega}\right)\right) h'_\xi\left(\frac{\gamma_k}{\omega}\right) w_k\,. \tag{15}$$

Depending on the oscillator and in turn the path h_ξ these integrals can be weakly singular (to be precise, having roots in the denominator) which can in turn be accounted for by the generalized Gauss-Laguerre quadrature. These singularities originate from roots involved in the multi-valued inverses of g at some stationary points.
For our problem we will find that we can always glue together the two opposite straight line paths at the single stationary point, apply one further transformation and finally use Gauss-Hermite quadrature. The whole process for computing (1) then looks like shown in Figure 5. Performing these transformations on the first two example wavepackets ϕ_k and ϕ_l from Figure 2 we can make the plot shown in 6.

Figure 5: Steepest descent transformation and integral computation.

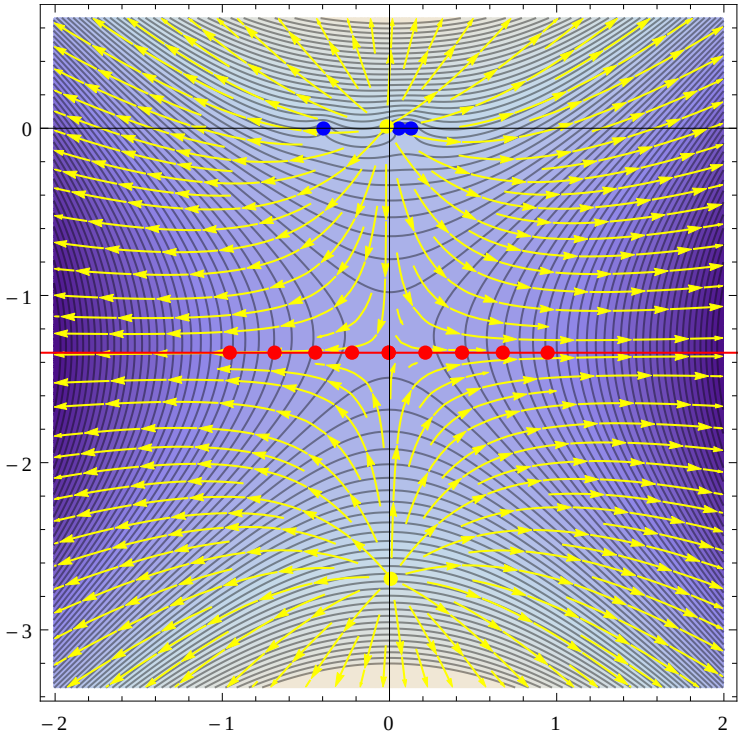


Figure 6: Contour levels and zeros (blue) of $|\overline{\phi_{\underline{l}}}\phi_{\underline{k}}|$, Gradient field and zeros of $|g|$ (yellow), Path of steepest descent with 9 Gauss Hermite quadrature nodes (red), in complex plane.

3.2 Extensions and new concepts

In the last part we have walked through the very basic concepts of the numerical steepest descent method. Under some broad assumptions this was proven to work for finite intervals and an arbitrary number of real stationary points [15]. The extension into higher dimensions has been worked out [16]. However, the state of the theory is too narrow for our setting. First we need to consider complex stationary points too. If such a point is near to the real line, it will influence the oscillator enough to become relevant. Figure 7 for example shows the integrand of:

$$\int_{-2}^{2} \left(2 + x^2\right) e^{50\imath\left(\delta x + \frac{x^3}{3}\right)} \mathrm{d}x \tag{16}$$

for different values of the parameter δ. One stationary point is located on the imaginary axis at $x^* = \imath\sqrt{\delta}$ and comes close to the real line. The influence is

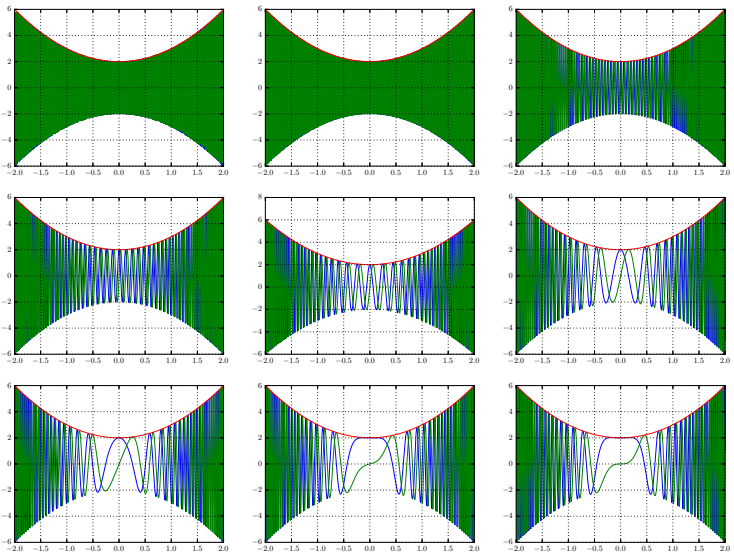


Figure 7: Oscillatory integral on $[-2, 2]$ with a stationary point x^* at $\imath\sqrt{\delta}$ shown for several different decreasing values of $\delta \in \{10, 5, 2, 1, \frac{1}{2}, \frac{1}{5}, \frac{1}{10}, \frac{1}{100}, 0\}$.

clearly present for small values of δ and soon becomes negligible for larger values. The general advise for complex stationary points is that the contour should pass through this point as shown in Figure 8, see section 4.3 in [15]. This will then give an exact decomposition in the sense of (14). It turns out this works very well for our setting. Actually, the *only* thing we have is a single stationary point. Another direction to extend the theory are infinite regions Ω. In our case we will need $\Omega = \mathbb{R}^N$. Before we take this step, we look at semi-finite intervals. For semi-finite intervals $[a, \infty[$ we attach only one single path $h_a(\tau)$ an the finite endpoint a. Details are described by Majidian and can be found in [17]. If there are stationary points, they can be split off into sub-problems over finite intervals. As an example, we look at the following integral:

$$I = \int_0^\infty \frac{100}{100 + x^2} \exp\left(\imath\omega e^{\sqrt{x}}\right) \mathrm{d}x \tag{17}$$

where we set $\omega = 2$. The path is given by the expression:

$$h_a(\tau) = \log(1 + \imath\tau)^2 \,. \tag{18}$$

After the change of variables the integral takes the following ugly form:

$$I = 200\, e^{\imath\omega} \int_0^\infty \frac{\imath \log(1 + \imath\tau) e^{-\omega\tau}}{(1 + \imath\tau)\left(100 + \log^4(1 + \imath\tau)\right)} \mathrm{d}\tau \,. \tag{19}$$

Given all these parts, the extension to infinite intervals becomes simple. We just split the overall interval $]-\infty, \infty[$ into a union of three intervals: $]-\infty, a]$, $[a, b]$

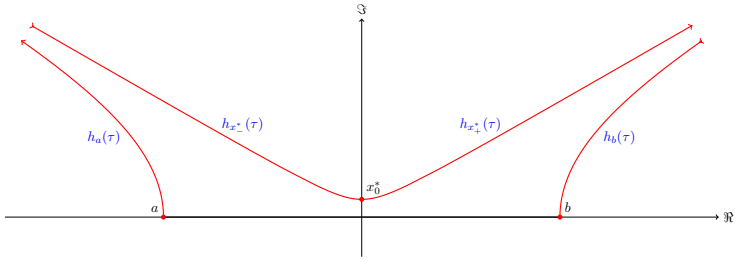
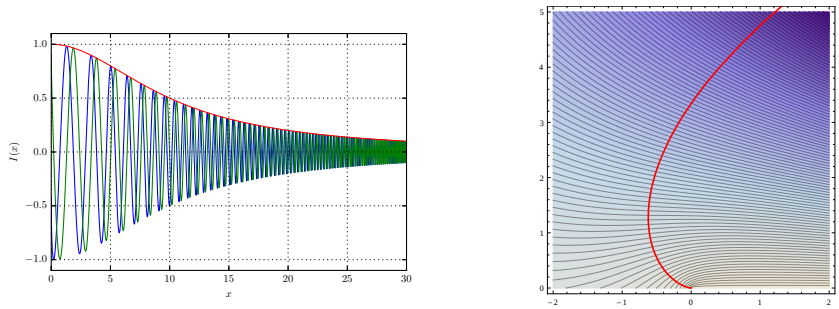


Figure 8: Paths of the integral (16). The two additional paths $h_{x^*,\pm}(\tau)$ go through the complex stationary point x_0^*. The contribution stemming from these two paths varies with δ. For $\delta = 1$ it is $\mathcal{O}\left(10^{-16}\right)$ and the whole integral has a value of $I = 0.0317496$ while for $\delta = \frac{1}{5}$ it is 0.033533 which is not negligible given that now $I = 0.0253591$.



(a) Oscillatory integrand on $[a, \infty[$.

(b) Path of steepest descend.

Figure 9: Example of an oscillatory integrand on a semi-finite interval and the corresponding path of steepest descend.

and $[b, \infty[$. The reason to choose three instead of only two semi-finite intervals is the stationary point which we know to exist in our specific case. With this splitting we can put it into the finite interval and know that it gets handled correctly. Of course the paths at the points a and b will be traversed two times in reversed directions and hence cancel out. The only thing that remains is the pair of paths attached to the single complex stationary point x^*. This works the same in any number of dimensions where we resolve the nested integrals one after the other. By gluing together the two paths of opposite direction we end up with a cross of exactly N straight paths. Therefore the number and nesting depth of integrals to perform does *not* grow compared to direct integration. This is of course very fortunate.

4 A first exemplary computation

In this section we show a complete computation for a three dimensional example. The purpose is to give the reader a rough feeling on how the overall process for our specific overlap integral works. We will write down all the technical details in the next section.
For the following computation we assume that the upper triangular matrix $\mathbf{T}$ is

given:

$$\mathbf{T} = \begin{pmatrix} t_{1,1} & t_{1,2} & t_{1,3} \\ 0 & t_{2,2} & t_{2,3} \\ 0 & 0 & t_{3,3} \end{pmatrix} \tag{20}$$

and we study the oscillator g given by:

$$g\left(\underline{x}\right) := \underline{x}^{\mathrm{H}} \mathbf{T} \underline{x}\,, \tag{21}$$

i.e.:

$$g\left(x_1, x_2, x_3\right) = t_{1,1}x_1^2 + t_{2,2}x_2^2 + t_{3,3}x_3^2 + t_{1,2}x_1x_2 + t_{1,3}x_1x_3 + t_{2,3}x_2x_3\,. \tag{22}$$

The integral we want to compute is:

$$I = \iiint_{-\infty}^{\infty} f\left(\underline{x}\right) \exp\left(i\omega g\left(\underline{x}\right)\right) \mathrm{d}\underline{x} = \iiint_{-\infty}^{\infty} i\left(x_1, x_2, x_3\right) \mathrm{d}x_1 \mathrm{d}x_2 \mathrm{d}x_3 \tag{23}$$

where we collapsed the internals into a single function $i\left(x_1, x_2, x_3\right)$ for reasons of notation. By using the expanded form of g the integral can be written as:

$$\begin{aligned} I = \iiint f\left(\underline{x}\right) &\exp\left(i\omega\left(t_{1,1}x_1^2 + t_{1,2}x_1x_2 + t_{1,3}x_1x_3\right)\right) \\ &\exp\left(i\omega\left(t_{2,2}x_2^2 + t_{2,3}x_2x_3\right)\right) \\ &\exp\left(i\omega\left(t_{3,3}x_3^2\right)\right) \mathrm{d}\underline{x} \\ = \iiint f\left(\underline{x}\right) &\exp\left(i\omega\left(t_{1,1}x_1^2 + t_{1,2}x_1x_2 + t_{1,3}x_1x_3\right)\right) \mathrm{d}x_1 \\ &\exp\left(i\omega\left(t_{2,2}x_2^2 + t_{2,3}x_2x_3\right)\right) \mathrm{d}x_2 \\ &\exp\left(i\omega\left(t_{3,3}x_3^2\right)\right) \mathrm{d}x_3\,. \end{aligned} \tag{24}$$

We pulled out of each integral the parts of the oscillator independent of the integration variable. Therefore we split the full oscillator g additively into N parts g_i:

$$g(\underline{x}) = g_1(x_1, x_2, x_3) + g_2(x_2, x_3) + g_3(x_3)\,. \tag{25}$$

Beginning with the inner most one, we can resolve these onion-like nested integrals one by one. From:

$$\int_{-\infty}^{\infty} f\left(\underline{x}\right) \exp\left(i\omega\left(t_{1,1}x_1^2 + t_{1,2}x_1x_2 + t_{1,3}x_1x_3\right)\right) \mathrm{d}x_1 \tag{26}$$

we find the relevant oscillator $g_1(x_1, x_2, x_3)$ to be:

$$g_1\left(x_1\right) := t_{1,1}x_1^2 + t_{1,2}x_1x_2 + t_{1,3}x_1x_3\,. \tag{27}$$

where we treat the variables x_2 and x_3 as parameters. Let x_1^* denote the (perhaps complex) stationary point[3] of g_1. The path equation then is:

$$g_1\left(h_1(\tau_1)\right) = g_1\left(x_1^*\right) + i\tau_1 \tag{28}$$

[3] A *stationary point* $\underline{x}^*$ is the solution to the equation $\nabla g(\underline{x}) = 0$, a point where the derivative of the oscillator vanishes. Actually, x_i^* is not a *point*. It depends on x_{i+1} up to x_N, so x_i^* is rather a $N - i$ dimensional object.

which, expanded out, is:

$$t_{1,1}h_1^2 + t_{1,2}h_1x_2 + t_{1,3}h_1x_3 = k_1(x_2, x_3) + i\tau_1 \tag{29}$$

$$t_{1,1}h_1^2 + t_{1,2}h_1x_2 + t_{1,3}h_1x_3 - k_1(x_2, x_3) - i\tau_1 = 0\,. \tag{30}$$

Since the expression $g_1\left(x_1^*\right)$ depends parametrically on x_2 and x_3, we denote the whole thing by $k_1\left(x_2, x_3\right)$. As k_1 does not depend on x_1 we can pull it out from the current integral. We will see later how to deal explicitly with all the k_i depending on all the variables $(x_{i+1}, \ldots, x_N)$.
By solving the last equation we obtain the complex integration path $h_1\left(\tau_1\right)$ parametrised in the variable $\tau_1 \in [0, \infty[$. Note that, at this point h_1 still depends on x_2 and x_3. Transforming the integrand of this first integral we find:

$$i_1\left(\tau_1, x_2, x_3\right) := f\left(h_1(\tau_1, x_2, x_3), x_2, x_3\right) \exp\left(i\omega k_1\left(x_2, x_3\right)\right) \exp\left(-\omega\tau_1\right) \frac{\partial h_1\left(\tau_1, x_2, x_3\right)}{\partial \tau_1}$$

and for the integral itself:

$$\begin{aligned} I_1\left(x_2, x_3\right) &= \int_0^\infty i_1\left(\tau_1, x_2, x_3\right) \mathrm{d}\tau_1 \\ &= \exp\left(i\omega\, k_1(x_2, x_3)\right) \int_0^\infty f\left(h_1(\tau_1, x_2, x_3), x_2, x_3\right) \exp\left(-\omega\tau_1\right) \frac{\partial h_1\left(\tau_1, x_2, x_3\right)}{\partial \tau_1} \mathrm{d}\tau_1\,. \end{aligned}$$

We pull out the prefactor $\exp\left(i\omega\, k_1(x_2, x_3)\right)$ constant with respect to x_1 and call it $p_1\left(x_2, x_3\right)$. Then we continue with the second integral where we have to merge parts of p_1 into g_2 (we will show the details later):

$$\int_{-\infty}^\infty I_1\left(x_2, x_3\right) \exp\left(i\omega\, k_1(x_2, x_3)\right) \exp\left(i\omega\left(t_{2,2}x_2^2 + t_{2,3}x_2x_3\right)\right) \mathrm{d}x_2\,. \tag{31}$$

The oscillator is given now by:

$$g_2(x_2) = t_{2,2}x_2^2 + t_{2,3}x_2x_3 + k_1(x_2, x_3) \tag{32}$$

and depends parametrically on x_3. For the path we have:

$$g_2\left(h_2(\tau_2)\right) = g_2\left(x_2^*\right) + i\tau_2 \tag{33}$$

which in turn can be resolved to:

$$t_{2,2}h_2^2 + t_{2,3}h_2x_3 + k_1(h_2, x_3) = k_2(x_3) + i\tau_2 \tag{34}$$

and then:

$$t_{2,2}h_2^2 + t_{2,3}h_2x_3 + k_1(h_2, x_3) - k_2(x_3) - i\tau_2 = 0\,. \tag{35}$$

Using this path h_2 obtained from solving the quadratic path equation we get the new integrand as:

$$i_2\left(\tau_2, x_3\right) := I_1\left(h_2(\tau_2, x_3), x_3\right) \exp\left(i\omega\, k_2(x_3)\right) \exp\left(-\omega\tau_2\right) \frac{\partial h_2\left(\tau_2, x_3\right)}{\partial \tau_2}\,. \tag{36}$$

And then in turn we write for the integral:

$$\begin{aligned} I_2\left(x_3\right) &= \int_0^\infty i_2\left(\tau_2, x_3\right) \mathrm{d}\tau_2 \\ &= \exp\left(i\omega\, k_2(x_3)\right) \int_0^\infty I_1\left(h_2(\tau_2, x_3), x_3\right) \exp\left(-\omega\tau_2\right) \frac{\partial h_2\left(\tau_2, x_3\right)}{\partial \tau_2} \mathrm{d}\tau_2\,. \end{aligned}$$

where we pull out $p_2 := \exp\left(i\omega k_2\left(x_3\right)\right)$. For the third and outer-most integral we can write:

$$\int_{-\infty}^{\infty} I_2\left(x_3\right) \exp\left(i\omega k_2\left(x_3\right)\right) \exp\left(i\omega t_{3,3} x_3^2\right) \mathrm{d}x_3 \,. \tag{37}$$

Once more, we first extract the oscillator expression:

$$g_3\left(x_3\right) = t_{3,3} x_3^2 + k_2(x_3) \,. \tag{38}$$

The corresponding path equation is:

$$\begin{aligned} g_3\left(h_3\left(\tau_3\right)\right) &= g_3(x_3^*) + i\tau_3 \\ t_{3,3} h_3^2 + k_2(h_3) &= k_3 + i\tau_3 \\ t_{3,3} h_3^2 + k_2(h_3) - k_3 - i\tau_3 &= 0 \,. \end{aligned} \tag{39}$$

Note that the variable k_3 does not depend on any x_i anymore. With the solution h_3 our integrand reads:

$$i_3\left(\tau_3\right) := I_2\left(h_3(\tau_3)\right) \exp\left(i\omega k_3\right) \exp\left(-\omega\tau_3\right) \frac{\partial h_3\left(\tau_3\right)}{\partial \tau_3} \tag{40}$$

and the integral becomes:

$$\begin{aligned} I_3 &= \int_0^{\infty} i_3\left(\tau_3\right) \mathrm{d}\tau_3 \\ &= \exp\left(i\omega k_3\right) \int_0^{\infty} I_2\left(h_3(\tau_3)\right) \exp\left(-\omega\tau_3\right) \frac{\partial h_3\left(\tau_3\right)}{\partial \tau_3} \mathrm{d}\tau_3 \,. \end{aligned} \tag{41}$$

Finally we obtain the solution of the original integral:

$$\begin{aligned} I = I_3 &= \int_0^{\infty} I_2\left(h_3(\tau_3)\right) \ldots \mathrm{d}\tau_3 \\ &= \int_0^{\infty} \int_0^{\infty} I_1\left(h_2(\tau_2, h_3(\tau_3)), h_3(\tau_3)\right) \ldots \mathrm{d}\tau_2 \ldots \mathrm{d}\tau_3 \\ &= \int_0^{\infty} \int_0^{\infty} \int_0^{\infty} f\left(h_1(\tau_1, h_2(\tau_2, h_3(\tau_3)), h_3(\tau_3)), h_2(\tau_2, h_3(\tau_3)), h_3(\tau_3)\right) \ldots \mathrm{d}\tau_1 \ldots \mathrm{d}\tau_2 \ldots \mathrm{d}\tau_3 \\ &= \int_0^{\infty} \int_0^{\infty} \int_0^{\infty} f\left(\tilde{h}_1(\tau_1, \tau_2, \tau_3), \tilde{h}_2(\tau_2, \tau_3), \tilde{h}_3(\tau_3)\right) \ldots \mathrm{d}\tau_1 \ldots \mathrm{d}\tau_2 \ldots \mathrm{d}\tau_3 \end{aligned}$$

This is the nasty return of the recursive scheme here. The final denested paths $\tilde{h}_1\left(\tau_1, \tau_2, \tau_3\right)$ and $\tilde{h}_2\left(\tau_2, \tau_3\right)$ can be found from $h_1\left(\tau_1, x_2, x_3\right)$ and $h_2\left(\tau_2, x_3\right)$ by substituting h_2 for x_2 and h_3 for x_3.

5 Systematic approach

In this section we look at the general N dimensional case and follow a very systematic approach. We will investigate all the technical details omitted during the example from the last section.

5.1 General triangular oscillators

Given a general, probably complex, upper triangular matrix $\mathbf{T}$ of dimension $N \times N$ we look at the oscillator given by:

$$g\left(\underline{x}\right) = \underline{x}^{\mathrm{H}} \mathbf{T} \underline{x}\,. \tag{42}$$

The process of pulling out from each integral parts that do not depend on the respective variable of integration yields the additive decomposition of g:

$$g\left(\underline{x}\right) = g\left(x_1, \ldots, x_N\right) = \sum_{i=1}^{N} g_i(x_i, \ldots, x_N)\,. \tag{43}$$

At each shell i of the onion we exclusively work with the partial oscillator g_i. In general the coefficients of this oscillator are given by the i-th row of $\mathbf{T}$:

$$\boxed{g_i\left(x_i, x_{i+1}, \ldots, x_N\right) := t_{i,i} x_i^2 + \sum_{j=i+1}^{N} t_{i,j} x_i x_j\,.} \tag{44}$$

5.2 Computing stationary points

As we know from the theory of numerical steepest descent, we need to find the stationary points of an oscillator g_i. It is possible to do this by a straight forward computation.
First we take the partial derivative of g_i, in fact we have:

$$\dot{g}_i := \frac{\partial g_i}{\partial x_i} = 2 t_{i,i} x_i + \sum_{j=i+1}^{N} t_{i,j} x_j \tag{45}$$

The location of the stationary point is implicitly specified by:

$$\dot{g}_i = 0\,. \tag{46}$$

Solving this equation for the stationary point x_i^* explicitly we get:

$$\boxed{x_i^* := x_i^*\left(x_{i+1}, \ldots, x_N\right) = -\frac{\sum_{j=i+1}^{N} t_{i,j} x_j}{2 t_{i,i}}\,.} \tag{47}$$

It is important to remember that at level i the point x_i^* is stationary with respect to the coordinate x_i but depends on the variables x_{i+1} through x_N.
Plugging this solution x_i^* into g_i from (44) we can compute the expressions k_i explicitly:

$$\begin{aligned} g_i(x_i^*, x_{i+1}, \ldots, x_N) &= t_{i,i} \left(-\frac{\sum_{j=i+1}^{N} t_{i,j} x_j}{2 t_{i,i}} \right)^2 + \sum_{j=i+1}^{N} t_{i,j} \left(-\frac{\sum_{k=i+1}^{N} t_{i,k} x_k}{2 t_{i,i}} \right) x_j \\ &= \frac{1}{4 t_{i,i}} \left(\sum_{j=i+1}^{N} t_{i,j} x_j \right)^2 - \frac{1}{2 t_{i,i}} \left(\sum_{j=i+1}^{N} t_{i,j} x_j \right) \left(\sum_{k=i+1}^{N} t_{i,k} x_k \right) \end{aligned}$$

and finally:

$$\boxed{k_i\left(x_{i+1},\ldots,x_N\right) := g_i\left(x_i^*, x_{i+1},\ldots,x_N\right) = -\frac{1}{4t_{i,i}}\left(\sum_{j=i+1}^{N} t_{i,j}x_j\right)^2 .} \tag{48}$$

5.3 Setting up and solving path equations

Central to this subsection is the path equation for g_i and how to solve it. We will use formula (44) in combination with (47) and (48) for this task. To begin with, the full path equation including all dependencies for the oscillator g_i and the corresponding path h_i reads:

$$g_i\left(h_i\left(p_i, x_{i+1},\ldots,x_N\right), x_{i+1},\ldots,x_N\right) = g_i\left(x_i^*\left(x_{i+1},\ldots,x_N\right), x_{i+1},\ldots,x_N\right) + ip_i\,.$$

Keeping only the most important dependencies we drop the dependence of g_i, h_i and x_i^* on x_j for $j > i$ and simply write:

$$g_i\left(h_i\left(p_i\right)\right) = g_i\left(x_i^*\right) + ip_i\,. \tag{49}$$

Expanding the left hand side by using (44) we find:

$$g_i\left(h_i\right) = t_{i,i}h_i^2 + \sum_{j=i+1}^{N} t_{i,j}h_i x_j = t_{i,i}h_i^2 + h_i\sum_{j=i+1}^{N} t_{i,j}x_j\,. \tag{50}$$

Using formula (48) for the right hand side and combining both sides one obtains:

$$\underbrace{t_{i,i}}_{A}h_i^2 + h_i\underbrace{\sum_{j=i+1}^{N} t_{i,j}x_j}_{B} + \underbrace{\frac{1}{4t_{i,i}}\left(\sum_{j=i+1}^{N} t_{i,j}x_j\right)^2 - ip_i}_{C} = 0\,. \tag{51}$$

What we got is a simple quadratic equation for the path h_i. Its solution is given by the well known explicit formula:

$$h_i = \frac{-B \pm \sqrt{\Delta}}{2A} \quad \text{where} \quad \Delta := B^2 - 4AC\,. \tag{52}$$

Now let's actually carry out the computation. The discriminant seems to be complicated at the first sight:

$$\begin{aligned}
\Delta = B^2 - 4AC &= \left(\sum_{j=i+1}^{N} t_{i,j}x_j\right)^2 - 4t_{i,i}\left(\frac{1}{4t_{i,i}}\left(\sum_{j=i+1}^{N} t_{i,j}x_j\right)^2 - ip_i\right) \\
&= \left(\sum_{j=i+1}^{N} t_{i,j}x_j\right)^2 - 4t_{i,i}\frac{1}{4t_{i,i}}\left(\sum_{j=i+1}^{N} t_{i,j}x_j\right)^2 + 4t_{i,i}ip_i \\
&= 4t_{i,i}ip_i
\end{aligned} \tag{53}$$

but actually it is surprisingly simple! For the complete path we then get:

$$h_i = \frac{-\sum_{j=i+1}^{N} t_{i,j}x_j \pm \sqrt{4t_{i,i}ip_i}}{2t_{i,i}} \tag{54}$$

which transforms into a nicer version:

$$\boxed{h_i\left(p_i, x_{i+1}, \ldots, x_N\right) = \pm\sqrt{\frac{ip_i}{t_{i,i}}} - \frac{1}{2t_{i,i}} \sum_{j=i+1}^{N} t_{i,j} x_j \,.} \tag{55}$$

It is important to note that the whole path depends only *linearly* on the remaining higher ranked variables x_j with $j > i$. As soon as $t_{i,j} = 0$ for all $j \neq i$ we get a bunch of completely decoupled paths.
In a next step we can now easily compute the derivative of any path:

$$\begin{aligned} \frac{\partial h_i}{\partial p_i} &= \frac{\partial}{\partial p_i}\left(\pm\sqrt{\frac{ip_i}{t_{i,i}}} - \frac{1}{2t_{i,i}} \sum_{j=i+1}^{N} t_{i,j} x_j\right) \\ &= \pm\frac{\partial}{\partial p_i}\sqrt{\frac{ip_i}{t_{i,i}}} \end{aligned} \tag{56}$$

and find this simple expression:

$$\boxed{\dot{h}_i := \frac{\partial h_i\left(p_i\right)}{\partial p_i} = \pm\frac{\sqrt{i}}{2\sqrt{t_{i,i}}\sqrt{p_i}} \,.} \tag{57}$$

It is obvious that for these formulae to be valid $t_{i,i}$ must never be zero. Formula (55) allows us now to compute explicitly starting with h_N all the paths h_{N-1}, h_{N-2} up to h_1 in this reversed order. For later usage we denote the paths resulting from this recursive resolution by $\tilde{h}_i$. An explicit description becomes very nasty soon:

$$\begin{aligned} \tilde{h}_N(p_N) &:= h_N(p_N) \\ \tilde{h}_{N-1}(p_{N-1}, p_N) &:= h_{N-1}(p_{N-1}, h_N(p_N)) \\ \tilde{h}_{N-2}(p_{N-2}, p_{N-1}, p_N) &:= h_{N-2}(p_{N-2}, h_{N-1}(p_{N-1}, h_N(p_N)), h_N(p_N)) \\ &\vdots \\ \tilde{h}_1(p_1, \ldots, p_N) &:= h_1(p_1, h_2(\ldots), \ldots, h_{N-1}(p_{N-1}, h_N(p_N)), h_N(p_N)) \end{aligned}$$

We can collect all the paths into a single vector like:

$$\underline{\tilde{h}}\left(\underline{p}\right) := \begin{bmatrix} \tilde{h}_1(p_1, p_2, \ldots, p_N) \\ \tilde{h}_2(p_2, \ldots, p_N) \\ \vdots \\ \tilde{h}_{N-1}(p_{N-1}, p_N) \\ \tilde{h}_N(p_N) \end{bmatrix} = \begin{bmatrix} \tilde{h}_1(p_1) \\ \tilde{h}_2(p_2) \\ \vdots \\ \tilde{h}_{N-1}(p_{N-1}) \\ \tilde{h}_N(p_N) \end{bmatrix} \tag{58}$$

The last relation is of course a notational shortcut and not a mathematical equality in general. For reasons which will become clear later we are in the end only interested in the paths with positive sign. If not specified otherwise, this is the implicit choice.
If we stack all the paths h_i into a vector $\underline{h}$, then this composition of all paths can be done easily as shown in Algorithm 1. By employing some matrix algebra, then the inner loop can even be avoided.

Algorithm 1 Procedure for composing the path

for all $i \in [N, N-1, \ldots, 1]$ **do**
  $h_i = \sqrt{\frac{ip_i}{t_{i,i}}}$
  for all $j \in [i+1, \ldots, N]$ **do**
    $h_i = h_i - \frac{t_{i,j}}{2t_{i,i}} h_j$
  end for
end for

5.4 Transforming the integrand

In the first step, we completely denest the integral from (1) as much as possible. We start with the inner-most integrand (note that we already have split the oscillator):

$$i_1(x_1, \ldots, x_N) := f(\underline{x}) \exp(i\omega g_1(x_1, \ldots, x_N)) \tag{59}$$

and compute its integral as:

$$I_1(x_2, \ldots, x_N) = \int_{-\infty}^{\infty} i_1(x_1, \ldots, x_N) \, \mathrm{d}x_1 \,. \tag{60}$$

Then we continue with the second shell:

$$\begin{aligned} i_2(x_2, \ldots, x_N) &:= I_1(x_2, \ldots, x_N) \exp(i\omega g_2(x_2, \ldots, x_N)) \\ I_2(x_3, \ldots, x_N) &= \int_{-\infty}^{\infty} i_2(x_2, \ldots, x_N) \, \mathrm{d}x_2 \,. \end{aligned} \tag{61}$$

This process continues recursively until no more integration variables are left:

$$\begin{aligned} i_N(x_N) &:= I_{N-1}(x_N) \exp(i\omega g_N(x_N)) \\ I_N() &= \int_{-\infty}^{\infty} i_N(x_N) \, \mathrm{d}x_N \,. \end{aligned} \tag{62}$$

and we obtain the final result $I \equiv I_N$. Each of the integrals I_i involved in this scheme is oscillatory with an oscillator g_i.
The main goal of this whole effort is to find a suitable transformation of variables $\underline{x} = (x_1, \ldots, x_N)$ into new variables $\underline{\tau} = (\tau_1, \ldots, \tau_N)$ and rewrite all the integrals above in a way such that they are no longer oscillatory.
For that purpose we computed the paths h_i which implement exactly this variable transformation. By construction, the real part of $g_i(h_i)$ is constant and for the composition we have:

$$g_i(h_i(p_i, x_{i+1}, \ldots, x_N)) = k_i(x_{i+1}, \ldots, x_N) + ip_i \,. \tag{63}$$

Plugging this into the exponential yields:

$$\begin{aligned} \exp(i\omega g_i(x_i, \ldots, x_N)) &= \exp(i\omega(k_i(x_{i+1}, \ldots, x_N) + ip_i)) \\ &= \exp(i\omega k_i(x_{i+1}, \ldots, x_N)) \exp(-\omega p_i) \end{aligned} \tag{64}$$

where the first term is still oscillatory but does not depend on x_i and the second term became exponentially decaying. Next, at stage i, we transform the whole integral I_i. Here we have to pay attention to the fact that formula (55) yields

two paths h_i^+ and h_i^- with different signs. From the general theory in [7] we know, that we have to use both and that the transformed integral I_i consists of two integrals, one for each path:

$$I_i[h_i] = I_i^{(1)}[h_i^+] - I_i^{(2)}[h_i^-] \tag{65}$$

Given the integrand $i_i(x_i, \ldots, x_N)$ we use the path h_i to make a variable transformation going from x_i to p_i. The first integral $I_i^{(1)}[h_i^+]$ therefore reads:

$$\begin{aligned} I_i^{(1)}[h_i^+]\left(x_{i+1}, \ldots, x_N\right) &= \exp\left(i\omega k_i\left(x_{i+1}, \ldots, x_N\right)\right) \\ &\cdot \int_0^\infty i_i\left(h_i^+\left(p_i\right), x_{i+1}, \ldots, x_N\right) \exp\left(-\omega p_i\right) \frac{\partial h_i^+\left(p_i\right)}{\partial p_i} \mathrm{d}p_i \end{aligned} \tag{66}$$

and we should not forget that the path h_i depends not only on p_i but also on $x_{i+1}, \ldots, x_N$. We will see in the next section how to handle the factor in front of the integral and therefore omit it in the following.
This integral above is singular for $p_i \to 0$ because of the square root of p_i in the denominator of the path derivative. We can fix this by the substitution $q = \sqrt{p}$, hence $p = q^2$ and $\mathrm{d}p = 2q\mathrm{d}q$. Applying this first to the path in (55) and the path derivative in (57) we obtain:

$$h_i\left(q_i, x_{i+1}, \ldots, x_N\right) = \pm\sqrt{\frac{\imath}{t_{i,i}}} q_i - \frac{1}{2t_{i,i}} \sum_{j=i+1}^{N} t_{i,j} x_j \tag{67}$$

and

$$\dot{h}_i := \frac{\partial h_i\left(q_i\right)}{\partial q_i} = \pm\sqrt{\frac{\imath}{t_{i,i}}} \frac{1}{2q_i}\,. \tag{68}$$

In the integral, the two factors of $2q_i$ cancel and we are left with:

$$I_i^{(1)}[h_i^+] = \int_0^\infty i_i\left(h_i^+\left(q_i\right), x_{i+1}, \ldots, x_N\right) \sqrt{\frac{\imath}{t_{i,i}}} \exp\left(-\omega q_i^2\right) \mathrm{d}q_i \tag{69}$$

The combination of the $\exp(q_i^2)$ factor and the integration range $[0, \infty[$ is a bit unfortunate for our goal of applying a quadrature. This combination corresponds to none of the classical Gaussian quadratures and one would have to build a custom rule for computing nodes and weights. Luckily, it turns out that, by the help of another variable transformation the path h_i^- can be transformed into h_i^+. The substitution is trivial and reads $r_i = -q_i$, therefore $q_i = -r_i$ and $\mathrm{d}q_i = -\mathrm{d}r_i$. We apply it in the second integral $I_i^{(2)}[h_i^-]$ only:

$$\begin{aligned} I_i^{(2)}[h_i^-] &= \int_0^\infty i_i\left(h_i^-\left(q_i\right), x_{i+1}, \ldots, x_N\right) \left(-\sqrt{\frac{\imath}{t_{i,i}}}\right) \exp\left(-\omega q_i^2\right) \mathrm{d}q_i \\ &= -\int_0^{-\infty} i_i\left(h_i^+\left(r_i\right), x_{i+1}, \ldots, x_N\right) \left(-\sqrt{\frac{\imath}{t_{i,i}}}\right) \exp\left(-\omega r_i^2\right) \mathrm{d}r_i \\ &= -\int_{-\infty}^0 i_i\left(h_i^+\left(r_i\right), x_{i+1}, \ldots, x_N\right) \sqrt{\frac{\imath}{t_{i,i}}} \exp\left(-\omega r_i^2\right) \mathrm{d}r_i \end{aligned} \tag{70}$$

At this point we can return to the complete integral I_i

$$\begin{aligned} I_i[h_i] &= I_i^{(1)}[h_i^+] - I_i^{(2)}[h_i^-] = I_i^{(1)}[h_i^+] + I_i^{(2)}[h_i^+] \\ &= \int_0^\infty i_i\left(h_i^+\left(q_i\right), x_{i+1}, \ldots, x_N\right) \sqrt{\frac{\imath}{t_{i,i}}} \exp\left(-\omega q_i^2\right) \mathrm{d}q_i \\ &+ \int_{-\infty}^0 i_i\left(h_i^+\left(r_i\right), x_{i+1}, \ldots, x_N\right) \sqrt{\frac{\imath}{t_{i,i}}} \exp\left(-\omega r_i^2\right) \mathrm{d}r_i \end{aligned} \tag{71}$$

and after gluing together the two parts, the final integral is:

$$I_i\left(x_{i+1}, \ldots, x_N\right) = \int_{-\infty}^\infty i_i\left(h_i^+\left(\tau_i\right), x_{i+1}, \ldots, x_N\right) \sqrt{\frac{\imath}{t_{i,i}}} \exp\left(-\omega \tau_i^2\right) \mathrm{d}\tau_i\,. \tag{72}$$

and we reached the goal. This transformed integral is no longer oscillatory and well suited for Gauss-Hermite quadrature. Applying the corresponding path transformation and variable substitutions to each of the I_i and putting together the results we find at the end of the day:

$$\boxed{I = \int_{-\infty}^\infty \cdots \int_{-\infty}^\infty f\left(\tilde{\underline{h}}(\underline{\tau})\right) \prod_{i=1}^N \sqrt{\frac{\imath}{t_{i,i}}} \exp\left(-\omega \tau_i^2\right) \mathrm{d}\tau_1 \cdots \mathrm{d}\tau_N} \tag{73}$$

This integral posses a quadratic exponential decay in all variables and hence can be computed easily by classical Gauss-Hermite quadrature rules.

5.5 Handling oscillatory prefactors

The prefactor term $P_i := \exp\left(i\omega k_i\left(x_{i+1}, \ldots, x_N\right)\right)$ appearing in the integral I_i is oscillatory. Hence the next enclosing integral is again of oscillatory type and therefore we have to merge the factor P_i with the next-level oscillator $g_{i+1}\left(x_{i+1}, \ldots, x_N\right)$. Mathematically, this means we compute an update $\grave{g}_{i+1}$ such that:

$$\grave{g}_{i+1}\left(x_{i+1}, \ldots, x_N\right) := g_{i+1}\left(x_{i+1}, \ldots, x_N\right) + k_i\left(x_{i+1}, \ldots, x_N\right)\,. \tag{74}$$

This seems to be easy, but it is not as straight forward as one might think. Remembering the general form of k_i shown in (48) we can start with:

$$\begin{aligned} k_i(x_{i+1}, \ldots, x_N) &= -\frac{1}{4t_{i,i}} \left(\sum_{j=i+1}^N t_{i,j} x_j \right)^2 \\ &= -\frac{1}{4t_{i,i}} \left(\sum_{j=i+1}^N t_{i,j}^2 x_j^2 + 2 \sum_{j=i+1}^N \sum_{k=j+1}^N t_{i,j} x_j t_{i,k} x_k \right)\,. \end{aligned} \tag{75}$$

The main question to answer next is which terms to keep and use for the updated oscillator $\grave{g}_{i+1}$ replacing g_{i+1} and which ones to move one level up outside this particular integral. The solution is pretty simple: we keep the x_{i+1}^2 and all of

$x_{i+1}x_j$. To achieve this we split the terms inside k_i into two groups:

$$k_i = -\frac{1}{4t_{i,i}}\left(\sum_{j=i+1}^{N} t_{i,j}^2 x_j^2 + 2\sum_{j=i+1}^{N}\sum_{k=j+1}^{N} t_{i,j}x_j t_{i,k}x_k\right)$$
$$= -\frac{1}{4t_{i,i}}\left(\underbrace{t_{i,i+1}^2 x_{i+1}^2}_{\text{keep}} + \underbrace{\sum_{j=i+2}^{N} t_{i,j}^2 x_j^2}_{\text{move}} + \underbrace{2\sum_{k=i+2}^{N} t_{i,i+1}x_{i+1}t_{i,k}x_k}_{\text{keep}} + \underbrace{2\sum_{j=i+2}^{N}\sum_{k=j+1}^{N} t_{i,j}x_j t_{i,k}x_k}_{\text{move}}\right).$$

Taking now formula (44) for g_{i+1} we can compute $\grave{y}_{i+1}$ from (74) as:

$$\grave{y}_{i+1} = t_{i+1,i+1}x_{i+1}^2 - \frac{1}{4t_{i,i}}t_{i,i+1}^2 x_{i+1}^2 + \sum_{j=i+2}^{N} t_{i+1,j}x_{i+1}x_j - \frac{1}{2t_{i,i}}\sum_{j=i+2}^{N} t_{i,i+1}x_{i+1}t_{i,j}x_j + \text{junk}$$
$$= \underbrace{\left(t_{i+1,i+1} - \frac{t_{i,i+1}^2}{4t_{i,i}}\right)}_{\grave{t}_{i+1,i+1}} x_{i+1}^2 + \sum_{j=i+2}^{N}\underbrace{\left(t_{i+1,j} - \frac{t_{i,i+1}t_{i,j}}{2t_{i,i}}\right)}_{\grave{t}_{i+1,j}} x_{i+1}x_j + \text{junk}\,.$$

All terms that we decided to move out are labeled as *junk* here and dealt with below. On the last line we see that we can write the partial oscillator $\grave{y}_{i+1}$ again in the standard form from (44), just the entries in the $i+1$-th row of the triangular matrix $\mathbf{T}$ have changed. We take into account the junk terms now. First notice that it is possible to split them once more into terms to keep and new junk terms:

$$-\frac{1}{4t_{i,i}}\left(\sum_{j=i+2}^{N} t_{i,j}^2 x_j^2 + 2\sum_{j=i+2}^{N}\sum_{k=j+1}^{N} t_{i,j}x_j t_{i,k}x_k\right)$$
$$= -\frac{1}{4t_{i,i}}\left(t_{i,i+2}^2 x_{i+2}^2 + \sum_{j=i+3}^{N} t_{i,j}^2 x_j^2 + 2\sum_{k=i+3}^{N} t_{i,i+2}x_{i+2}t_{i,k}x_k + 2\sum_{j=i+3}^{N}\sum_{k=j+1}^{N} t_{i,j}x_j t_{i,k}x_k\right).$$

The first and third term are in turn added to the oscillator $\grave{y}_{i+2}$. All others are handled recursively and added to $\grave{y}_k$ with $k \geq i+3$ until we reach $k = N$ and nothing is left. At that point we updated all rows of $\mathbf{T}$ below the i-th one.
The general formulae to accomplish this update for a fixed i and with $j > i$ and $k > j$ read:

$$\boxed{\begin{aligned}\grave{t}_{j,j} &:= t_{j,j} - \frac{t_{i,j}^2}{4t_{i,i}} \\ \grave{t}_{j,k} &:= t_{j,k} - \frac{t_{i,j}t_{i,k}}{2t_{i,i}}\end{aligned}} \tag{76}$$

and allow us to easily compute the new matrix $\mathbf{T_{i+1}}$ from $\mathbf{T_i}$ given at step i.
We have to repeat the whole procedure on the next outer shell $i+1$, handling the factor p_{i+1}.
Starting from $\mathbf{T_1} \equiv \mathbf{T}$ we compute one after the other $\mathbf{T_2}$, $\mathbf{T_3}$ until $\mathbf{T_N}$ which is the final oscillator matrix. The following Figures 10, 11 and 12 show the flow of information during this step-wise updates for a schematic matrix.

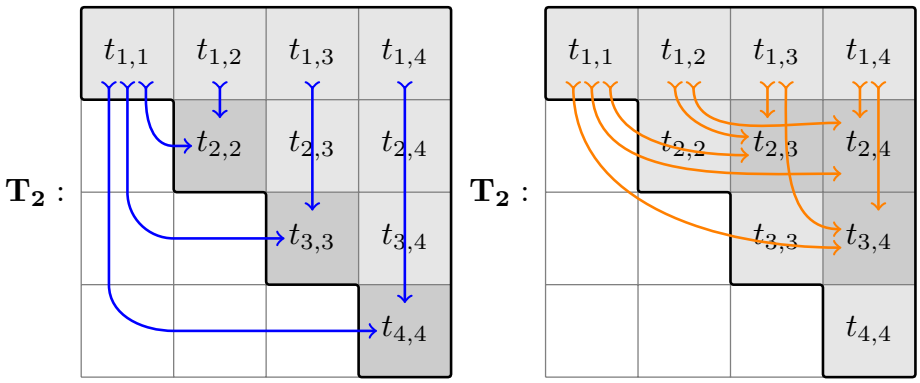


Figure 10: Oscillator matrix update going from $\mathbf{T_1}$ to $\mathbf{T_2}$

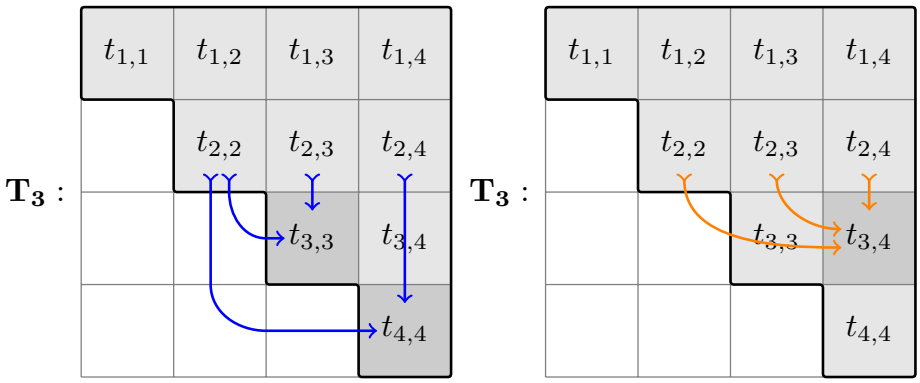


Figure 11: Oscillator matrix update going from $\mathbf{T_2}$ to $\mathbf{T_3}$

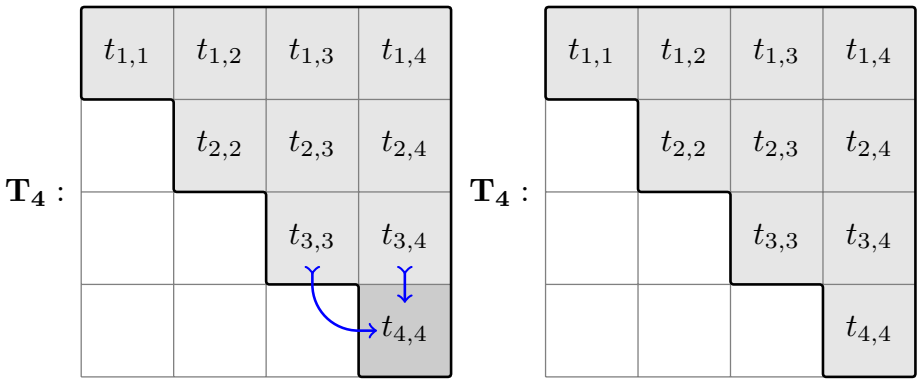


Figure 12: Oscillator matrix update going from $\mathbf{T_3}$ to $\mathbf{T_4}$

This whole procedure is written out in pseudo code in Algorithm 2 below. For this to work the diagonal matrix elements $\mathbf{T}_{i,i}$ must never become zero. Otherwise we would divide by zero. We can interpret this failure as a case where the oscillator is not quadratic anymore but behaves as a conic degenerated along at least one direction.

Algorithm 2 Procedure for updating the oscillator matrix $\mathbf{T}$

```
for all i ∈ [2, 3, ..., N] do
                                          ▷ Diagonal elements
    for all j ∈ [i, i+1, ..., N] do
        T_{j,j} := T_{j,j} − T²_{i,j} / (4 T_{i,i})
    end for
                                          ▷ Upper triangular elements
    for all r ∈ [i, i+1, ..., N] do
        for all c ∈ [r+1, r+2, ..., N] do
            T_{r,c} := T_{r,c} − T_{i,r} T_{i,c} / (2 T_{i,i})
        end for
    end for
end for
```

It is important to mention that we can do this transformation on $\mathbf{T}$ at the very beginning and before doing any other computations shown in this chapter.

5.6 Quadrature and quadrature rules

The quadrature rules necessary for evaluating the integral in (73) are the well known classical Gauss-Hermite rules. The Gauss-Hermite quadrature is built to approximate integrals of the form:

$$\int_{-\infty}^{\infty} f(x) \exp(-x^2) \mathrm{d}x \approx \sum_{k=1}^{n} f\left(\underline{\gamma}_k\right) \underline{w}_k \,. \tag{77}$$

In one dimension the quadrature nodes $\underline{\gamma}$ of a rule of order n are given by the roots of the Hermite polynomial $H_n(x)$ which is defined as:

$$H_n(x) := \sum_{i=0}^{\lfloor \frac{n}{2} \rfloor} \frac{(-1)^i n!}{i!(n-2i)!} (2x)^{n-2i} \tag{78}$$

The weights are given by the expression:

$$\underline{w}_k := \frac{2^{n-1} n! \sqrt{\pi}}{n^2 \left(H_{n-1}(x_i)\right)^2} \tag{79}$$

For multi-dimensional quadrature we can use a tensor product ansatz:

$$\underline{X} = \bigotimes_{i=1}^{N} \underline{x_i} \qquad \underline{W} = \bigotimes_{i=1}^{N} \underline{w_i} \,. \tag{80}$$

Another possibility would be to use sparse Smolyak grids. Looking again at the full dimensional integral from (73):

$$I = \int_{-\infty}^{\infty} \cdots \int_{-\infty}^{\infty} f\left(\tilde{\underline{h}}(\underline{\tau})\right) \prod_{i=1}^{N} \sqrt{\frac{\imath}{t_{i,i}}} \exp\left(-\omega \tau_i^2\right) \mathrm{d}\tau_1 \cdots \mathrm{d}\tau_N \tag{81}$$

we first remove the ω from all the exponential functions:

$$I = \frac{1}{\omega^{\frac{N}{2}}} \int_{-\infty}^{\infty} \cdots \int_{-\infty}^{\infty} f\left(\tilde{h}_1\left(\frac{\tau_1}{\sqrt{\omega}}\right), \ldots, \tilde{h}_N\left(\frac{\tau_N}{\sqrt{\omega}}\right)\right) \prod_{i=1}^{N} \exp\left(-\tau_i^2\right) \cdots \mathrm{d}\tau_1 \cdots \mathrm{d}\tau_N$$

and then apply the standard rule to find $Q \approx I$. At the end of the day, the quadrature we will use to approximate I looks like:

$$\boxed{Q := \prod_{j=1}^{N} \sqrt{\frac{\imath}{\omega t_{j,j}}} \sum_{k_1}^{n} \cdots \sum_{k_N}^{n} f\left(\tilde{h}_1\left(\frac{x_{k_1}}{\sqrt{\omega}}\right), \ldots, \tilde{h}_N\left(\frac{x_{k_N}}{\sqrt{\omega}}\right)\right) \prod_{i=1}^{N} w_{k_i}\,.} \tag{82}$$

6 General quadratic oscillator

In this section we consider a more general oscillator term than the one from equation (42). This time we try to solve the full quadratic problem:

$$g\left(\underline{x}\right) := \underline{x}^{\mathrm{H}} \mathbf{A} \underline{x} + \underline{b}^{\mathrm{H}} \underline{x} + c \tag{83}$$

where we do not make any assumptions on $\mathbf{A} \in \mathbb{C}^{N \times N}$. The general strategy we follow is to reduce this problem back to the triangular oscillator we studied in much detail and know how to solve.

6.1 Removing the linear term

First we want to get rid of the linear term $\underline{b}^{\mathrm{H}} \underline{x}$. This is done by a technique similar to completion of the square. Starting from the above oscillator we want to find a transformation χ that annihilates the linear term.
Assuming that this transformation is linear:

$$\underline{x}' = \chi^{-1}\left(\underline{x}\right) := \underline{x} - \underline{u} \quad \leftrightarrow \quad \underline{x} = \chi\left(\underline{x}'\right) = \underline{x}' + \underline{u}\,. \tag{84}$$

we find that:

$$\begin{aligned} \underline{x}'^{\mathrm{H}} \mathbf{A} \underline{x}' &= \left(\underline{x} - \underline{u}\right)^{\mathrm{H}} \mathbf{A} \left(\underline{x} - \underline{u}\right) \\ &= \underline{x}^{\mathrm{H}} \mathbf{A} \underline{x} - \underline{x}^{\mathrm{H}} \mathbf{A} \underline{u} - \underline{u}^{\mathrm{H}} \mathbf{A} \underline{x} + \underline{u}^{\mathrm{H}} \mathbf{A} \underline{u} \\ &= \underline{x}^{\mathrm{H}} \mathbf{A} \underline{x} - \underline{u}^{\mathrm{H}} \mathbf{A}^{\mathrm{H}} \underline{x} - \underline{u}^{\mathrm{H}} \mathbf{A} \underline{x} + \underline{u}^{\mathrm{H}} \mathbf{A} \underline{u}\,. \end{aligned} \tag{85}$$

If we match this against the definition of g we get the relations shown next. We are interested in the linear terms only for finding $\underline{u}$:

$$\begin{aligned} \underline{b}^{\mathrm{H}} \underline{x} &= -\underline{u}^{\mathrm{H}} \mathbf{A}^{\mathrm{H}} \underline{x} - \underline{u}^{\mathrm{H}} \mathbf{A} \underline{x} \\ \underline{b}^{\mathrm{H}} &= -\underline{u}^{\mathrm{H}} \left(\mathbf{A} + \mathbf{A}^{\mathrm{H}}\right) \\ \underline{b} &= -\left(\mathbf{A} + \mathbf{A}^{\mathrm{H}}\right) \underline{u} \end{aligned} \tag{86}$$

and finally:

$$\underline{u} = -\left(\mathbf{A} + \mathbf{A}^{\mathrm{H}}\right)^{-1}\underline{b}\,. \tag{87}$$

With the explicit value of u we can rewrite our χ in a final form:

$$\begin{aligned} \underline{x}' = \chi^{-1}(\underline{x}) &:= \underline{x} + \left(\mathbf{A} + \mathbf{A}^{\mathrm{H}}\right)^{-1}\underline{b} \\ \leftrightarrow \quad \underline{x} = \chi(\underline{x}') &= \underline{x}' - \left(\mathbf{A} + \mathbf{A}^{\mathrm{H}}\right)^{-1}\underline{b}\,. \end{aligned} \tag{88}$$

Then we can start computing the new oscillator g':

$$\begin{aligned} g(\underline{x}) &= g(\chi(\underline{x}')) \\ &= \left(\underline{x}' - \left(\mathbf{A} + \mathbf{A}^{\mathrm{H}}\right)^{-1}\underline{b}\right)^{\mathrm{H}}\mathbf{A}\left(\underline{x}' - \left(\mathbf{A} + \mathbf{A}^{\mathrm{H}}\right)^{-1}\underline{b}\right) + \underline{b}^{\mathrm{H}}\left(\underline{x}' - \left(\mathbf{A} + \mathbf{A}^{\mathrm{H}}\right)^{-1}\underline{b}\right) + c \end{aligned}$$

we expand and simplify this step by step:

$$\begin{aligned} &= \underline{x}'^{\mathrm{H}}\mathbf{A}\underline{x}' - \underline{x}'^{\mathrm{H}}\mathbf{A}\left(\mathbf{A} + \mathbf{A}^{\mathrm{H}}\right)^{-1}\underline{b} - \left(\left(\mathbf{A} + \mathbf{A}^{\mathrm{H}}\right)^{-1}\underline{b}\right)^{\mathrm{H}}\mathbf{A}\underline{x}' \\ &\quad + \left(\left(\mathbf{A} + \mathbf{A}^{\mathrm{H}}\right)^{-1}\underline{b}\right)^{\mathrm{H}}\mathbf{A}\left(\mathbf{A} + \mathbf{A}^{\mathrm{H}}\right)^{-1}\underline{b} + \underline{b}^{\mathrm{H}}\left(\underline{x}' - \left(\mathbf{A} + \mathbf{A}^{\mathrm{H}}\right)^{-1}\underline{b}\right) + c \\ &= \underline{x}'^{\mathrm{H}}\mathbf{A}\underline{x}' - \underline{x}'^{\mathrm{H}}\mathbf{A}\left(\mathbf{A} + \mathbf{A}^{\mathrm{H}}\right)^{-1}\underline{b} - \underline{b}^{\mathrm{H}}\left(\mathbf{A} + \mathbf{A}^{\mathrm{H}}\right)^{-\mathrm{H}}\mathbf{A}\underline{x}' \\ &\quad + \underline{b}^{\mathrm{H}}\left(\mathbf{A} + \mathbf{A}^{\mathrm{H}}\right)^{-\mathrm{H}}\mathbf{A}\left(\mathbf{A} + \mathbf{A}^{\mathrm{H}}\right)^{-1}\underline{b} + \underline{b}^{\mathrm{H}}\underline{x}' - \underline{b}^{\mathrm{H}}\left(\mathbf{A} + \mathbf{A}^{\mathrm{H}}\right)^{-1}\underline{b} + c\,. \end{aligned}$$

In a next step, try to unify the linear terms having only one $\underline{x}'$:

$$\begin{aligned} &\quad - \underline{x}'^{\mathrm{H}}\mathbf{A}\left(\mathbf{A} + \mathbf{A}^{\mathrm{H}}\right)^{-1}\underline{b} - \underline{b}^{\mathrm{H}}\left(\mathbf{A} + \mathbf{A}^{\mathrm{H}}\right)^{-\mathrm{H}}\mathbf{A}\underline{x}' + \underline{b}^{\mathrm{H}}\underline{x}' \\ &= -\underline{b}^{\mathrm{H}}\left(\mathbf{A} + \mathbf{A}^{\mathrm{H}}\right)^{-\mathrm{H}}\mathbf{A}^{\mathrm{H}}\underline{x}' - \underline{b}^{\mathrm{H}}\left(\mathbf{A} + \mathbf{A}^{\mathrm{H}}\right)^{-\mathrm{H}}\mathbf{A}\underline{x}' + \underline{b}^{\mathrm{H}}\underline{x}' \\ &= -\underline{b}^{\mathrm{H}}\left(\left(\mathbf{A} + \mathbf{A}^{\mathrm{H}}\right)^{-1}\mathbf{A}^{\mathrm{H}} + \left(\mathbf{A} + \mathbf{A}^{\mathrm{H}}\right)^{-1}\mathbf{A}\right)\underline{x}' + \underline{b}^{\mathrm{H}}\underline{x}' \\ &= -\underline{b}^{\mathrm{H}}\left(\mathbf{A} + \mathbf{A}^{\mathrm{H}}\right)^{-1}\left(\mathbf{A} + \mathbf{A}^{\mathrm{H}}\right)\underline{x}' + \underline{b}^{\mathrm{H}}\underline{x}' \\ &= -\underline{b}^{\mathrm{H}}\underline{x}' + \underline{b}^{\mathrm{H}}\underline{x}' = 0\,. \end{aligned}$$

and indeed we find that they vanish. This confirms that the transformation is correct. At the end of the day we obtain the new oscillator free of any linear term:

$$g'(\underline{x}') := \underline{x}'^{\mathrm{H}}\mathbf{A}\underline{x}' + \underline{b}^{\mathrm{H}}\left(\mathbf{A} + \mathbf{A}^{\mathrm{H}}\right)^{-\mathrm{H}}\mathbf{A}\left(\mathbf{A} + \mathbf{A}^{\mathrm{H}}\right)^{-1}\underline{b} - \underline{b}^{\mathrm{H}}\left(\mathbf{A} + \mathbf{A}^{\mathrm{H}}\right)^{-1}\underline{b} + c \tag{89}$$

where we can pack all the constant terms into the definition of c':

$$\begin{aligned} c' &:= \underline{b}^{\mathrm{H}}\left(\mathbf{A} + \mathbf{A}^{\mathrm{H}}\right)^{-\mathrm{H}}\mathbf{A}\left(\mathbf{A} + \mathbf{A}^{\mathrm{H}}\right)^{-1}\underline{b} - \underline{b}^{\mathrm{H}}\left(\mathbf{A} + \mathbf{A}^{\mathrm{H}}\right)^{-1}\underline{b} + c \\ &= -\frac{1}{2}\underline{b}^{\mathrm{H}}\left(\mathbf{A} + \mathbf{A}^{\mathrm{H}}\right)^{-1}\underline{b} + c\,. \end{aligned} \tag{90}$$

Therefore:

$$g'(\underline{x}') = \underline{x}'^{\mathrm{H}}\mathbf{A}\underline{x}' + c' \tag{91}$$

6.2 Triangularize the oscillator

Given an oscillator like (91) above with an arbitrary matrix $\mathbf{A}$ of full rank. We seek a coordinate transformation ρ that makes the matrix $\mathbf{A}$ (upper) triagonal. This is achieved by the Schur decomposition [11]:

$$\mathbf{A} = \mathbf{U}^{\mathrm{H}}\mathbf{T}\mathbf{U} \tag{92}$$

where:

$$\mathbf{T} = \begin{pmatrix} t_{1,1} & \dots & t_{1,N} \\ & \ddots & \vdots \\ 0 & & t_{N,N} \end{pmatrix} \tag{93}$$

and the matrix $\mathbf{U}$ is even unitary. (The decomposition is not necessarily unique but this does not matter for our purpose.) The variable transformation is then given by the following mapping:

$$\underline{x}'' = \rho^{-1}\left(\underline{x}'\right) := \mathbf{U}\underline{x}' \quad \leftrightarrow \quad \underline{x}' = \rho\left(\underline{x}''\right) = \mathbf{U}^{-1}\underline{x}'' = \mathbf{U}^{\mathrm{H}}\underline{x}'' \,. \tag{94}$$

For the oscillator we compute:

$$\begin{aligned} g'\left(\underline{x}'\right) &= g'\left(\rho\left(\underline{x}''\right)\right) = g'\left(\mathbf{U}^{\mathrm{H}}\underline{x}''\right) \\ &= \left(\mathbf{U}^{\mathrm{H}}\underline{x}''\right)^{\mathrm{H}}\mathbf{A}\left(\mathbf{U}^{\mathrm{H}}\underline{x}''\right) + c' \\ &= \underline{x}''^{\mathrm{H}}\mathbf{U}\mathbf{A}\mathbf{U}^{\mathrm{H}}\underline{x}'' + c' \\ &= \underline{x}''^{\mathrm{H}}\mathbf{T}\underline{x}'' + c' \end{aligned} \tag{95}$$

with $\mathbf{T}$ being upper triagonal. Finally we obtain the triangular oscillator:

$$g''\left(\underline{x}''\right) := \underline{x}''^{\mathrm{H}}\mathbf{T}\underline{x}'' + c' \,. \tag{96}$$

Combining both transformations χ and ρ into a single one we get the following pullback:

$$g\left(\underline{x}\right) = g\left(\chi\left(\rho\left(\underline{x}''\right)\right)\right) = g\left(\zeta\left(\underline{x}''\right)\right) \tag{97}$$

The related transformation ζ is the explicit composition of χ and ρ:

$$\begin{aligned} \underline{x}'' = \zeta^{-1}\left(\underline{x}\right) = \rho^{-1}\left(\chi^{-1}\left(\underline{x}\right)\right) = \rho^{-1}\left(\underline{x} + \left(\mathbf{A} + \mathbf{A}^{\mathrm{H}}\right)^{-1}\underline{b}\right) = \mathbf{U}\underline{x} + \mathbf{U}\left(\mathbf{A} + \mathbf{A}^{\mathrm{H}}\right)^{-1}\underline{b} \\ \underline{x} = \zeta\left(\underline{x}''\right) = \chi\left(\rho\left(\underline{x}''\right)\right) = \chi\left(\mathbf{U}^{\mathrm{H}}\underline{x}''\right) = \mathbf{U}^{\mathrm{H}}\underline{x}'' - \left(\mathbf{A} + \mathbf{A}^{\mathrm{H}}\right)^{-1}\underline{b} \,. \end{aligned}$$

With the help of ζ we can transform any oscillator of the form (83) into a triangular one like shown in (42).
In this variable transformation we need also to include the Jacobi determinant. Writing the first transformation as in (88) we have:

$$J_\chi = \frac{\partial\chi\left(\underline{x}'\right)}{\partial\underline{x}'} = \mathbb{1} \tag{98}$$

and hence $\det(J_\chi) = 1$. For the other transformation (94) it holds:

$$J_\rho = \frac{\partial\rho\left(\underline{x}''\right)}{\partial\underline{x}''} = \mathbf{U}^{-1} \tag{99}$$

from which we find $\det(J_\rho) = \det(\mathbf{U}^{-1}) = \frac{1}{\det\mathbf{U}}$. Because $\mathbf{U}$ is unitary we finally have $|\det\mathbf{U}| = 1$ and no additional factors appear in the integrals.

7 Some open issues

There are some yet unanswered questions in theory as well as in practical use. In this section we will briefly review them, find the main difficulties in each one and try to propose possible solutions.

We did not give a proof for the methods proposed. In the case of $\langle\phi_k|\phi_l\rangle$ and even $\langle\phi_k|V|\phi_l\rangle$ for some class of well-behaved, essentially polynomial, potential $V(x)$ there is nothing to prove. We rely on an intelligent change of variables to obtain a tame non-oscillatory integral. Only the harder case for general $V(x)$ is still left open. Despite comprehensive numerical evidence, we would like a full proof showing the validity of our extension to the theory of numerical steepest descent to infinite intervals.[4]

The next question is about the correct handling of stationary points since our single stationary point x^* is in general complex. There are some comments on that situation in the original reference [15]. The central observation made there is that the integration path should pass through the stationary point even if it is abroad in the complex plane. In our implementation we shift the whole oscillator such that $x^* = 0$. Having this point is essential for the reason that we need to attach the paths of steepest descent there.

One serious issue arises when we try to compute the integral $\langle\phi_k|\phi_l\rangle$ for large values of the indices k or l. These indices correspond directly to the degree of the polynomial part of any wavepacket ϕ. The Hermite polynomials stay in the envelope part f and are not treated by numerical steepest descent which targets only the oscillator g explicitly. In case of high degrees, the polynomials themselves become rapidly oscillatory and we end up with another oscillatory integral, this time *not* of a form like (1) where a steepest descend transformation is applicable. Figure 13 shows a typical example of such a polynomial oscillatory integrand.

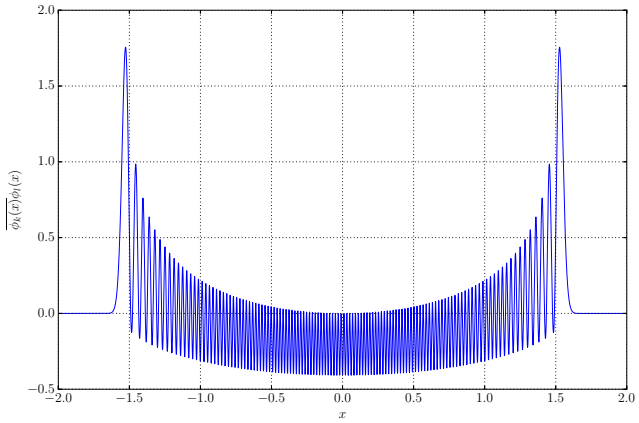


Figure 13: Integrand of the overlap of ϕ_{120} and ϕ_{122} with $\varepsilon = 0.1$. The parameter set $\Pi = \{0, 0, 1, \imath\}$ is identical for both packets and hence the exponential parts cancel to $\exp(-\frac{x^2}{\varepsilon^2})$. This example shows the oscillatory structure caused solely by the high degree of the polynomials involved.

In the end, the values k and l set the lower bound on the number of Gauss-

[4]The idea was to use the work from [15] on finite intervals and from [17] on semi-finite intervals to construct a solution for infinite intervals by gluing together three regions, one finite and two semi-finite ones. It turned out that the proof given in the latter reference is flawed. As is stands now we can not even assume the semi-finite part to be correct.

Hermite quadrature nodes that we have to use for correct integration. This is very unfortunate and can diminish the gain in number of nodes obtained through the steepest descent transformation. However, with modern algorithms it is easily possible to compute Gaussian quadrature with hundreds or even thousands of quadrature points [10, 20]. This makes the problem accessible for direct computation. Some work in this direction will be shown in a future report. One might try to resolve these oscillations by another method on top of what we did so far. Rewriting the Hermite polynomials into their integral formulation

$$\mathrm{H}_n(x) = \frac{\exp\left(\frac{3\pi\imath n}{2}\right) 2^n}{\sqrt{\pi}} \int_{-\infty}^{\infty} t^n e^{-(t-\imath x)^2} \mathrm{d}t$$

does not solve the problem as there is still a troublesome factor t^n. If we would apply the steepest descent transformation to this integral, computing the n-th power of the integration paths will again yield polynomials of high order.
The next open problem is about integrals $\langle\phi|\hat{o}|\phi\rangle$ including additional scalar multiplicative or differential operators $\hat{o}$. An example of this kind is the computation of expectation values of potential[5] energies: $\langle\phi|V|\phi\rangle$. In this example given, the same wavepacket ϕ appears in both, the bra and the ket part of the integral. Therefore the integration is much easier, oscillations caused by the exponential parts cancel out to a large degree. But let us consider the general case where we have $\langle\phi|\hat{o}|\phi'\rangle$. The main issue here are the possible effects on the oscillator g induced by $\hat{o}$. For example, assume that $\hat{o}$ is a potential $V(x)$. Then, V becomes part of the envelope and there apply some growth limits on the envelope f for $|x| \to \infty$. If V contains exponential parts, they have to be merged into the oscillator g and in turn change the integration paths. Further, if there are any poles in f, we have to be very careful with integration too and make sure to include all the relevant residuals. In the computations done so far, the envelope f is assumed to have no poles in $\mathbb{C}^N$. Altogether this poses some challenges for non-polynomial potentials. Polynomial potentials are tame enough to fit the framework shown.
Although there are some loose ends, we have constructed a method which works better for larger ω hence for smaller semiclassical scaling ε. It gives excellent results in the interesting and complicated case where $\Pi \approx \Pi'$.

8 Numerical Examples

8.1 Motivation

The initial motivation for this research originates from the computation of autocorrelations of wavepackets Ψ like:

$$I = \langle \Psi[\Pi] \,|\, \Psi[\Pi'] \rangle = \int \cdots \int_{\mathbb{R}^D} \overline{\Psi[\Pi]} \Psi[\Pi'] \mathrm{d}\underline{x} \,. \tag{100}$$

This turns out to be very challenging. In some of the most interesting cases where $\Pi \approx \Pi'$ holds or where the semiclassical scaling parameter ε becomes

[5] Kinetic energy expectation $\langle\phi|\hat{T}|\phi\rangle$ is in our case no issue because the gradient of a wavepacket can be expressed in a linear combination of new wavepackets again, hence we are back at the case $\langle\phi_k|\phi_l\rangle$.

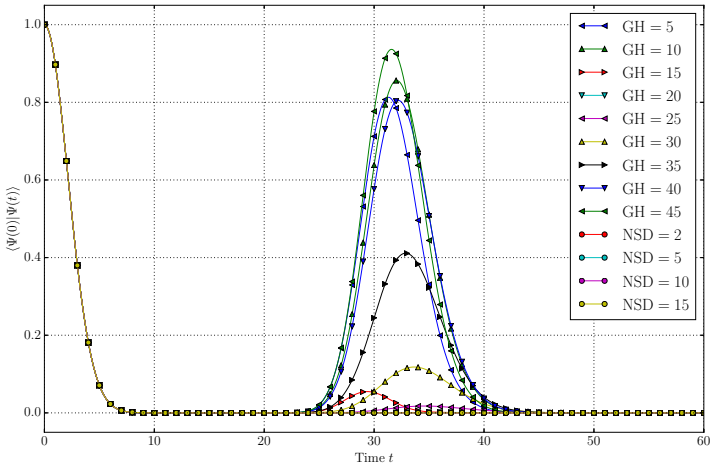


Figure 14: Autocorrelation of the Mercury dynamics computed with classical Gauss-Hermite quadrature and with the steepest descent method each with several numbers of quadrature points. While Gauss-Hermite quadrature fails, the numerical steepest descent transformation yields accurate results.

small, Gaussian quadrature introduces huge errors and hence we find spurious non-physical correlations.
For example we studied the time-dependent Schrödinger equation with a wavepacket Ψ propagating in a Morse potential $V(x)$. The potential was fit to experimental data which are taken from [19]. The semiclassical scaling parameter $\varepsilon = 0.048360$ representing mass ratio between nuclei and electrons matches the nuclear dynamics of the mercury compound Hg_2.
Gauss-Hermite quadrature with any number of nodes shows high spurious autocorrelation bumps, compare to Figure 14. Using the steepest descent transformation before applying a quadrature scheme gives correct results for an even smaller number of nodes.
In the following we will perform a number of different numerical simulations and show the robustness as well as pleasant convergence properties of this new technique. All simulations shown here were carried out by our implementation in the `WaveBlocks` simulation code [6].

8.2 Two-Packet Experiment

In this section we show the insufficiency of the Gauss-Hermite quadrature for computing the integral in (100). The setup of this experiment consists of two wavepackets $\phi[\Pi]$ and $\phi[\Pi']$. Both are fixed in space at positions q and q'. Next we direct the momenta p and p' towards each other ($p' = -p$) and start increasing their magnitude $|p|$. (Note that for all figures below, the values of the p axis are multiplicative factors on top of the wavepacket' original values p and p'.) The procedure is shown also in Figure 15.
The higher the momenta, the more oscillations appear in the product $\overline{\phi}(x)\phi'(x)$. These oscillations are difficult for Gauss Hermite quadrature to catch and accuracy will break down even for relatively small momenta. On the other hand,

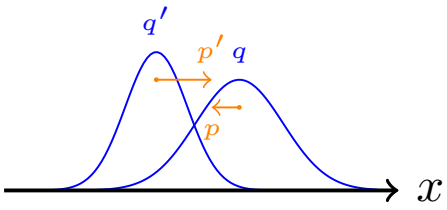
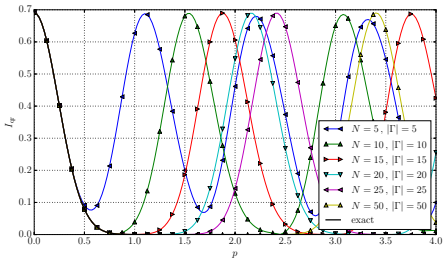


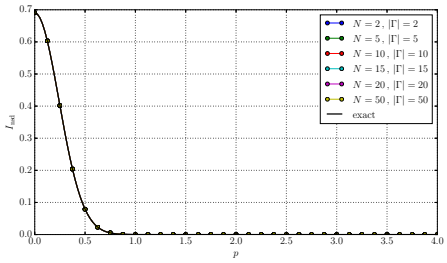
Figure 15: Setup of the first experiment. There are two wavepackets Ψ and Ψ' located next to each other at fixed positions q and q'. We set increasing momenta p and p' in opposite direction.

since the steepest descent transformation applies to arbitrary high oscillator frequency, it perfectly handles arbitrary momenta and converges fast to the correct overlap integral value.
For the one-dimensional case there exists an analytic formula for computing the integral for any state ϕ_k. Evaluation is very expensive but can nevertheless serve as exact reference solution. For examples in higher dimensions we can take the formula for general Gaussian integrals in case we look at ground states $\phi_{\underline{0}}$ only. Otherwise we take the computation including the steepest descent transformation using the largest number of quadrature nodes as reference.
The whole process of computing the integral works as shown in the Figure 5. Especially in the multi-dimensional case we still need a full tensor product of N Gauss-Hermite quadrature nodes in the end. This results in N^D quadrature nodes, which scales exponentially with the dimension D. The important bit is however that the value of N is *much* smaller compared to direct application of traditional quadrature schemes as done in 1. The only oscillations that remain are caused by the degree of the polynomial part of the integrand for large state index k. There, the order bounds of Gauss quadrature of course still apply.

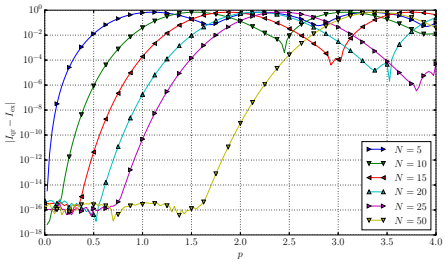
8.2.1 Convergence in $|p|$



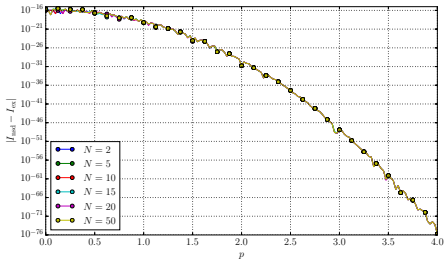
(a) Direct Gauss-Hermite quadrature of size N with a total of $|\Gamma|$ nodes.



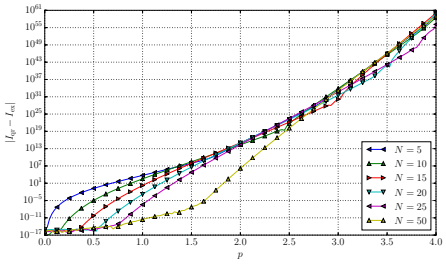
(b) Steepest descent transformation and Gauss-Hermite quadrature of size N with a total of $|\Gamma|$ nodes.



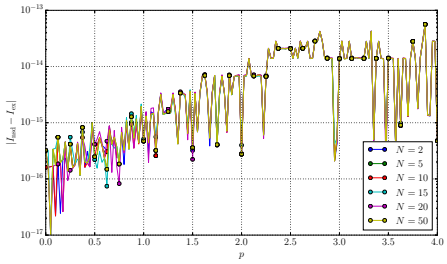
(c) Absolute error of the direct quadrature method compared to the exact solution.



(d) Absolute error of the steepest descent method compared to the exact solution.

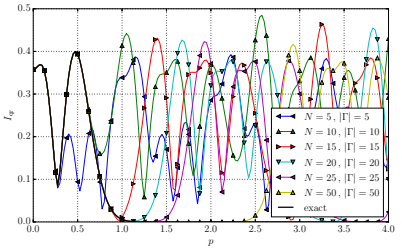


(e) Relative error of the direct quadrature method compared to the exact solution.

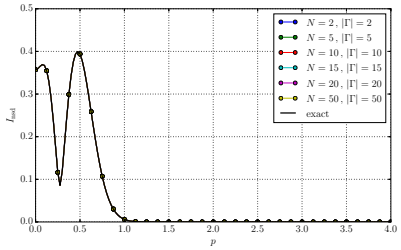


(f) Relative error of the steepest descent method compared to the exact solution.

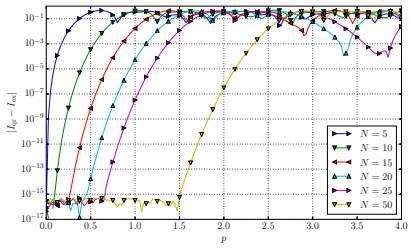
Figure 16: Experiment with ϕ_0 and ϕ_0'. The parameters are: $q = -0.2$, $p = 1.5$, $Q = 1.0$, $P = 1.0\imath$ and $q' = 0.125$, $p' = -1.5$, $Q' = 0.8$, $P' = 1.25\imath$ with $\varepsilon = 0.3$.



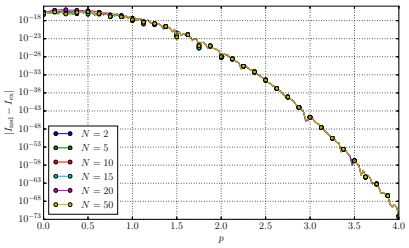
(a) Direct Gauss-Hermite quadrature of size N with a total of $|\Gamma|$ nodes.



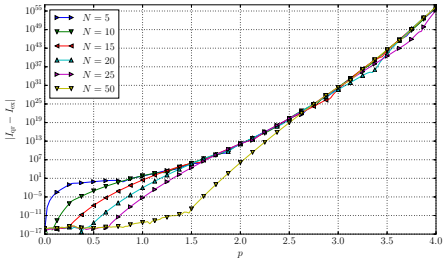
(b) Steepest descent transformation and Gauss-Hermite quadrature of size N with a total of $|\Gamma|$ nodes.



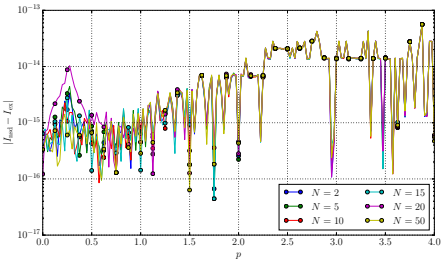
(c) Absolute error of the direct quadrature method compared to the exact solution.



(d) Absolute error of the steepest descent method compared to the exact solution.

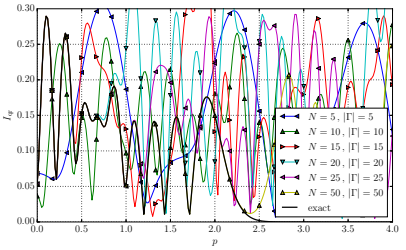


(e) Relative error of the direct quadrature method compared to the exact solution.

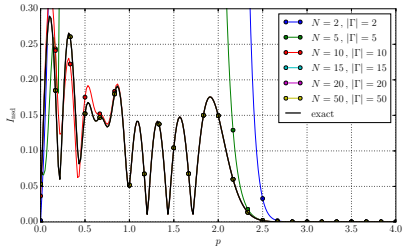


(f) Relative error of the steepest descent method compared to the exact solution.

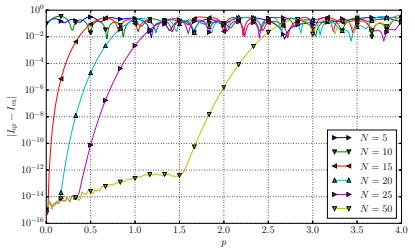
Figure 17: Experiment with ϕ_2 and ϕ_1'. The parameters are: $q = -0.2$, $p = 1.5$, $Q = 1.0$, $P = 1.0\imath$ and $q' = 0.125$, $p' = -1.5$, $Q' = 0.8$, $P' = 1.25\imath$ with $\varepsilon = 0.3$.



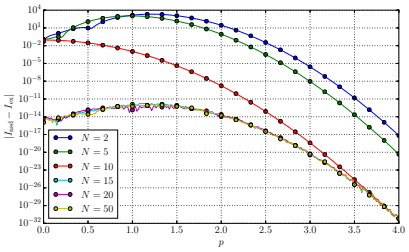
(a) Direct Gauss-Hermite quadrature of size N with a total of $|\Gamma|$ nodes.



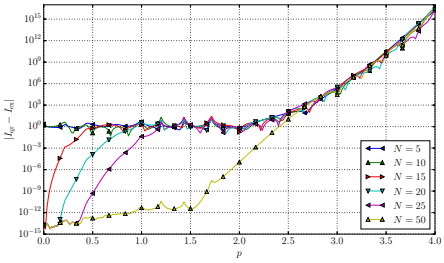
(b) Steepest descent transformation and Gauss-Hermite quadrature of size N with a total of $|\Gamma|$ nodes.



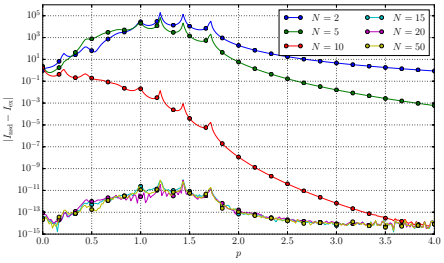
(c) Absolute error of the direct quadrature method compared to the exact solution.



(d) Absolute error of the steepest descent method compared to the exact solution.

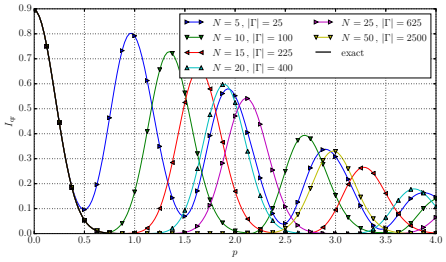


(e) Relative error of the direct quadrature method compared to the exact solution.

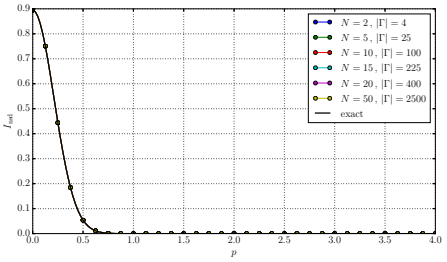


(f) Relative error of the steepest descent method compared to the exact solution.

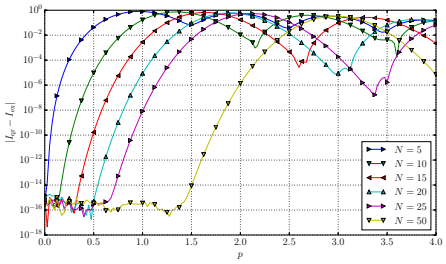
Figure 18: Experiment with ϕ_{11} and ϕ'_9. The parameters are: $q = -0.2$, $p = 1.2$, $Q = 1.0$, $P = 1.0\imath$ and $q' = 0.2$, $p' = -1.2$, $Q' = 0.5$, $P' = 2.0\imath$ with $\varepsilon = 0.3$.



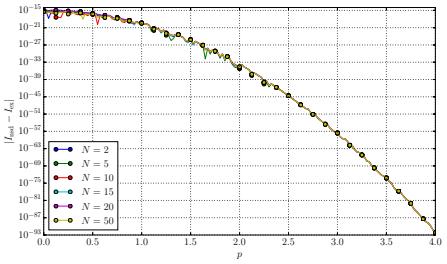
(a) Direct tensor product Gauss-Hermite quadrature of linear size N with a total of $|\Gamma|$ nodes.



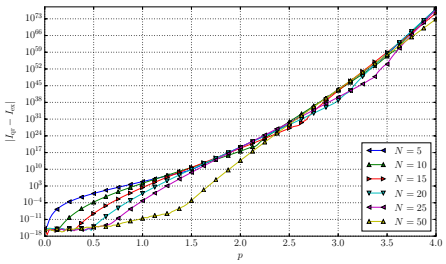
(b) Steepest descent transformation and tensor product Gauss-Hermite quadrature of linear size N with a total of $|\Gamma|$ nodes.



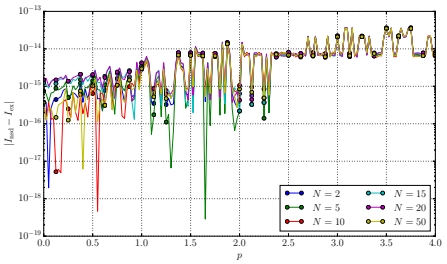
(c) Absolute error of the direct quadrature method compared to the exact solution.



(d) Absolute error of the steepest descent method compared to the exact solution.

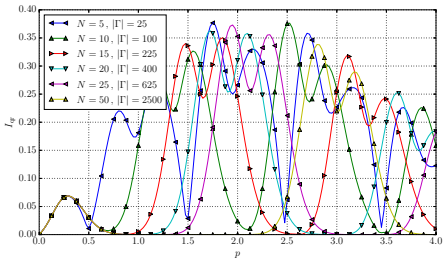


(e) Relative error of the direct quadrature method compared to the exact solution.

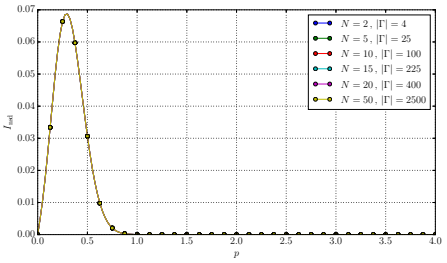


(f) Relative error of the steepest descent method compared to the exact solution.

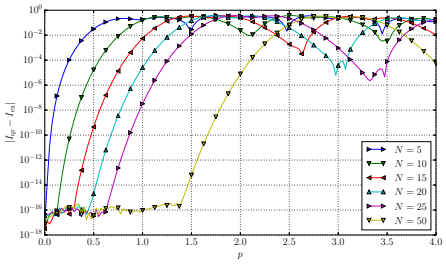
Figure 19: Experiment with $\phi_{0,0}$ and $\phi'_{0,0}$. The parameters are: $\underline{q} = (-0.1, 0.1)$, $\underline{p} = (1.0, -0.1)$, $\mathbf{Q} = \mathbf{1}$, $\mathbf{P} = \imath\mathbf{1}$ and $\underline{q}' = (0.1, 0.1)$, $\underline{p}' = (-1.0, 0.1)$, $\mathbf{Q}' = \mathbf{1}$, $\mathbf{P}' = \imath\mathbf{1}$ with $\varepsilon = 0.3$.



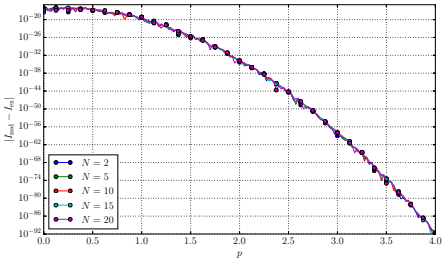
(a) Direct tensor product Gauss-Hermite quadrature of linear size N with a total of $|\Gamma|$ nodes.



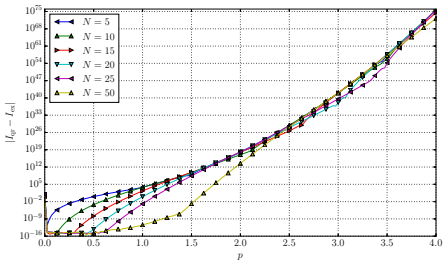
(b) Steepest descent transformation and tensor product Gauss-Hermite quadrature of linear size N with a total of $|\Gamma|$ nodes.



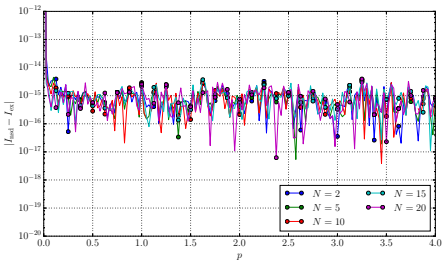
(c) Absolute error of the direct quadrature method compared to the exact solution.



(d) Absolute error of the steepest descent method compared to the exact solution.

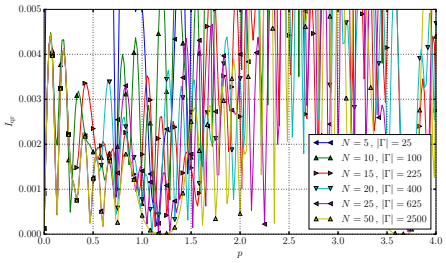


(e) Relative error of the direct quadrature method compared to the exact solution.

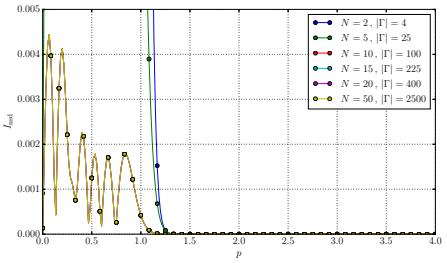


(f) Relative error of the steepest descent method compared to the exact solution.

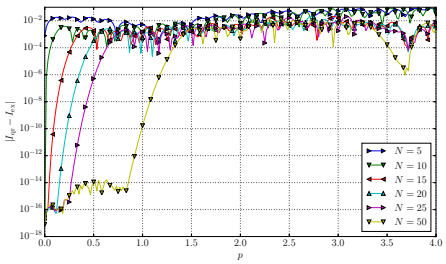
Figure 20: Experiment with $\phi_{1,0}$ and $\phi'_{0,1}$. The parameters are: $\underline{q} = (-0.1, 0.1)$, $\underline{p} = (1.0, -0.1)$, $\mathbf{Q} = \mathbf{1}$, $\mathbf{P} = \imath\mathbf{1}$ and $\underline{q}' = (0.1, 0.1)$, $\underline{p}' = (-1.0, 0.1)$, $\mathbf{Q}' = \mathbf{1}$, $\mathbf{P}' = \imath\mathbf{1}$ with $\varepsilon = 0.3$.



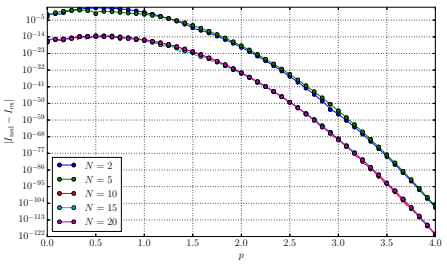
(a) Direct tensor product Gauss-Hermite quadrature of linear size N with a total of $|\Gamma|$ nodes.



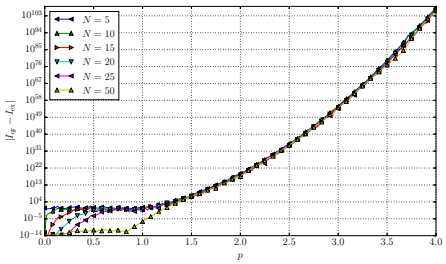
(b) Steepest descent transformation and tensor product Gauss-Hermite quadrature of linear size N with a total of $|\Gamma|$ nodes.



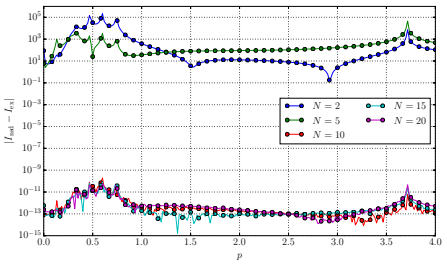
(c) Absolute error of the direct quadrature method compared to the exact solution.



(d) Absolute error of the steepest descent method compared to the exact solution.

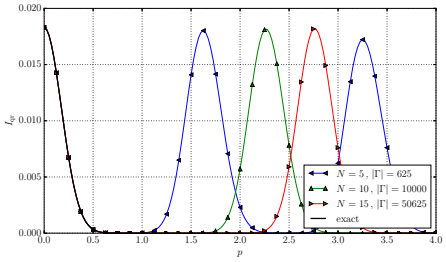


(e) Relative error of the direct quadrature method compared to the exact solution.

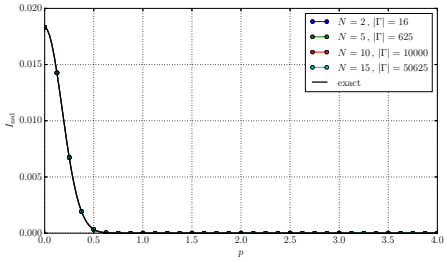


(f) Relative error of the steepest descent method compared to the exact solution.

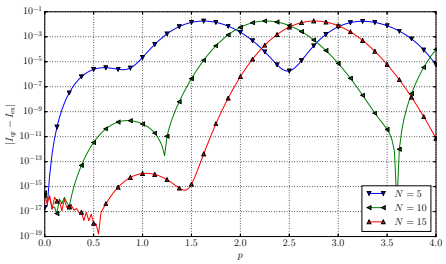
Figure 21: Experiment with $\phi_{8,8}$ and $\phi'_{8,8}$. The parameters are: $\underline{q} = (-0.1, 0.1)$, $\underline{p} = (1.0, -0.1)$, $\mathbf{Q} = (1.0, 0; 0, 1.0)$, $\mathbf{P} = (1.0\imath, 0; 0, 1.0\imath)$ and $\underline{q}' = (0.1, 0.1)$, $\underline{p}' = (-1.0, 0.1)$, $\mathbf{Q}' = (2.0, 0; 0, 0.5)$, $\mathbf{P}' = (0.5\imath, 0; 0, 2.0\imath)$ with $\varepsilon = 0.3$.



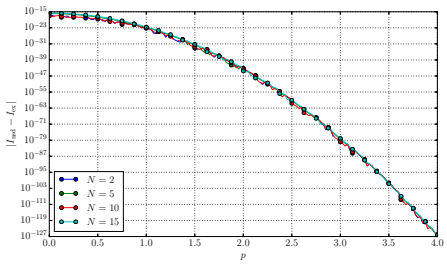
(a) Direct tensor product Gauss-Hermite quadrature of linear size N with a total of $|\Gamma|$ nodes.



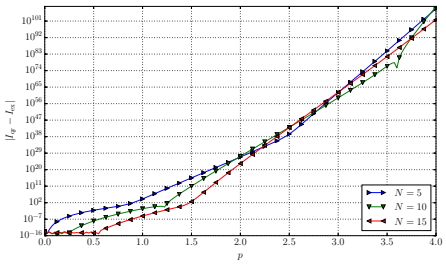
(b) Steepest descent transformation and tensor product Gauss-Hermite quadrature of linear size N with a total of $|\Gamma|$ nodes.



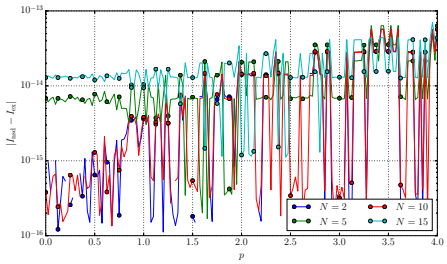
(c) Absolute error of the direct quadrature method compared to the exact solution.



(d) Absolute error of the steepest descent method compared to the exact solution.

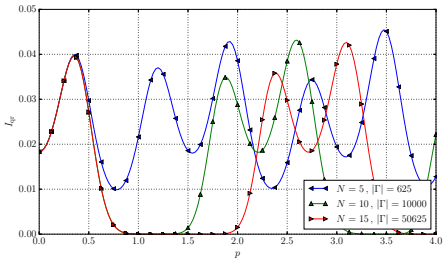


(e) Relative error of the direct quadrature method compared to the exact solution.

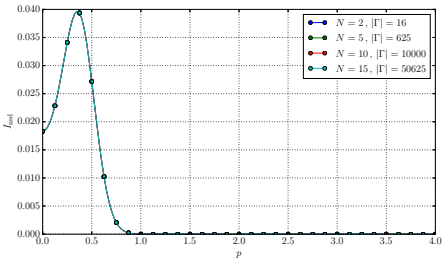


(f) Relative error of the steepest descent method compared to the exact solution.

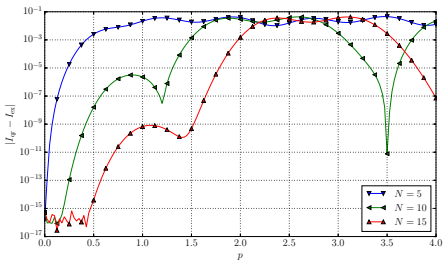
Figure 22: Experiment with $\phi_{0,0,0,0}$ and $\phi'_{0,0,0,0}$. The parameters are: $\underline{q} = (-0.1, -0.1, -0.1, -0.1)$, $\underline{p} = (0.2, -0.2, -0.2, 0.2)$, $\mathbf{Q} = \mathbf{1}$, $\mathbf{P} = \imath\mathbf{1}$ and $\underline{q'} = (0.1, 0.1, 0.1, 0.1)$, $\underline{p}' = (-0.2, 0.2, 0.2, -0.2)$, $\mathbf{Q}' = \mathbf{1}$, $\mathbf{P}' = \imath\mathbf{1}$ with $\varepsilon = 0.1$.



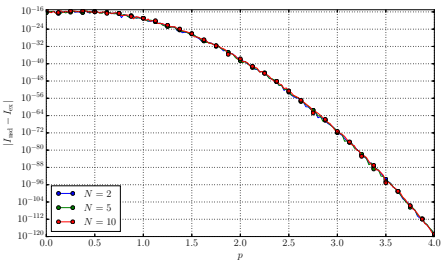
(a) Direct tensor product Gauss-Hermite quadrature of linear size N with a total of $|\Gamma|$ nodes.



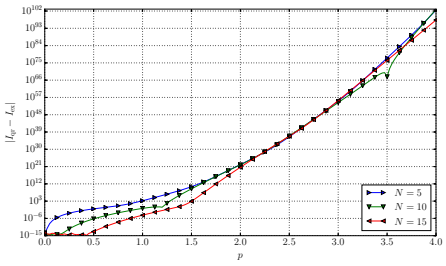
(b) Steepest descent transformation and tensor product Gauss-Hermite quadrature of linear size N with a total of $|\Gamma|$ nodes.



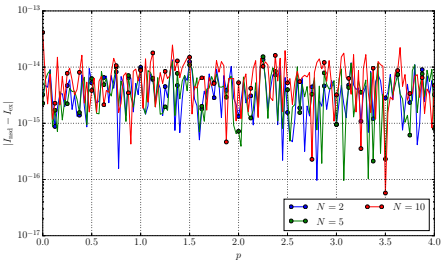
(c) Absolute error of the direct quadrature method compared to the exact solution.



(d) Absolute error of the steepest descent method compared to the exact solution.



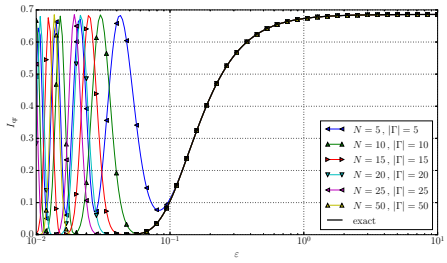
(e) Relative error of the direct quadrature method compared to the exact solution.



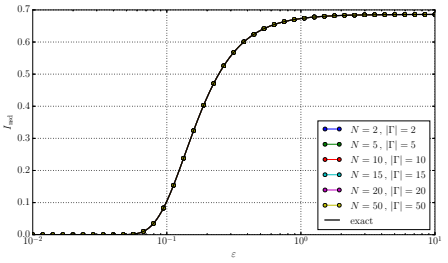
(f) Relative error of the steepest descent method compared to the exact solution.

Figure 23: Experiment with $\phi_{1,1,1,1}$ and $\phi'_{1,1,1,1}$. The parameters are: $\underline{q} = (-0.1, -0.1, -0.1, -0.1)$, $\underline{p} = (0.2, -0.2, -0.2, 0.2)$, $\mathbf{Q} = \mathbf{1}$, $\mathbf{P} = \imath\mathbf{1}$ and $\underline{q'} = (0.1, 0.1, 0.1, 0.1)$, $\underline{p}' = (-0.2, 0.2, 0.2, -0.2)$, $\mathbf{Q}' = \mathbf{1}$, $\mathbf{P}' = \imath\mathbf{1}$ with $\varepsilon = 0.1$.

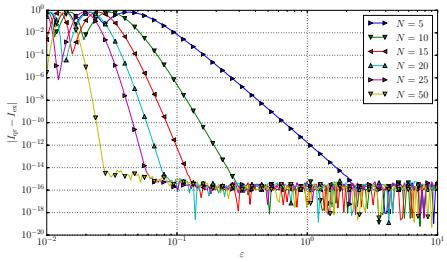
8.2.2 Convergence in ε



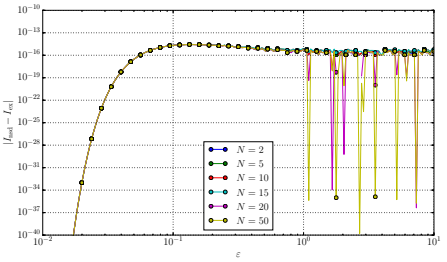
(a) Direct Gauss-Hermite quadrature of size N with a total of $|\Gamma|$ nodes.



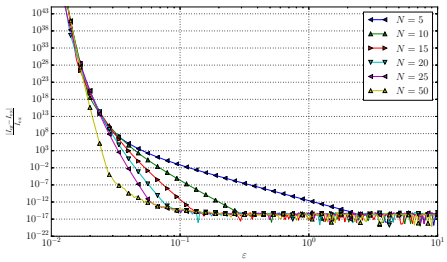
(b) Steepest descent transformation and Gauss-Hermite quadrature of size N with a total of $|\Gamma|$ nodes.



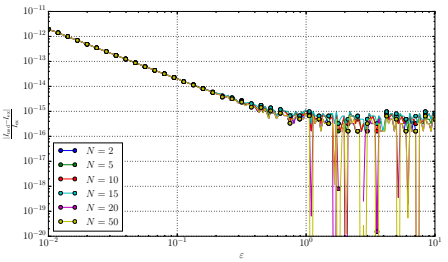
(c) Absolute error of the direct quadrature method compared to the exact solution.



(d) Absolute error of the steepest descent method compared to the exact solution.

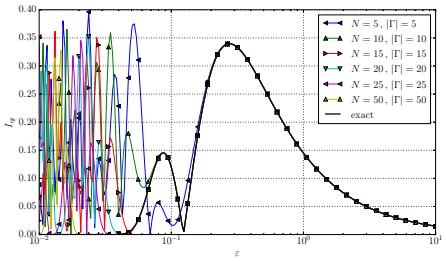


(e) Relative error of the direct quadrature method compared to the exact solution.

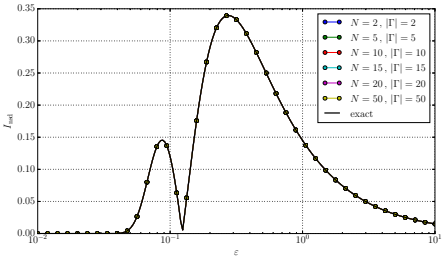


(f) Relative error of the steepest descent method compared to the exact solution.

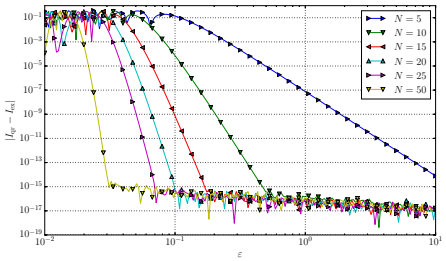
Figure 24: Experiment with ϕ_0 and ϕ_0'. The parameters are: $q = 1.0$, $p = 0.2$, $Q = 0.5$, $P = 2.0\imath$ and $q' = 1.0$, $p' = -0.2$, $Q' = 2.0$, $P' = 0.5\imath$.



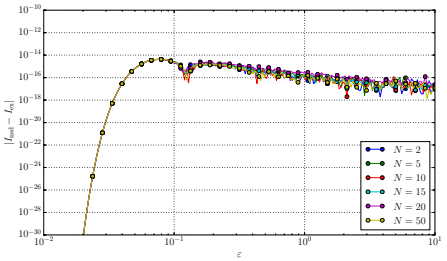
(a) Direct Gauss-Hermite quadrature of size N with a total of $|\Gamma|$ nodes.



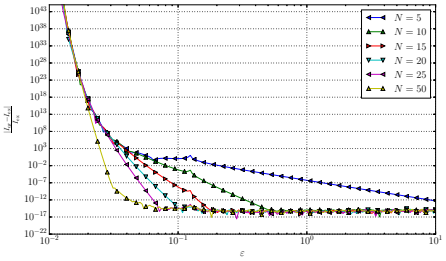
(b) Steepest descent transformation and Gauss-Hermite quadrature of size N with a total of $|\Gamma|$ nodes.



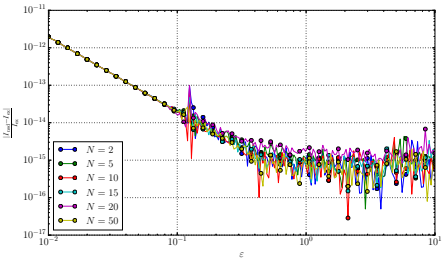
(c) Absolute error of the direct quadrature method compared to the exact solution.



(d) Absolute error of the steepest descent method compared to the exact solution.

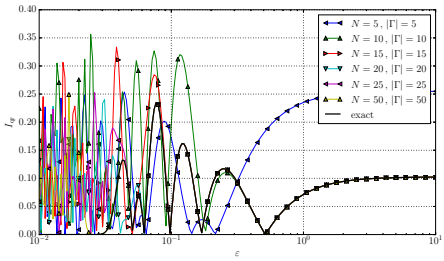


(e) Relative error of the direct quadrature method compared to the exact solution.

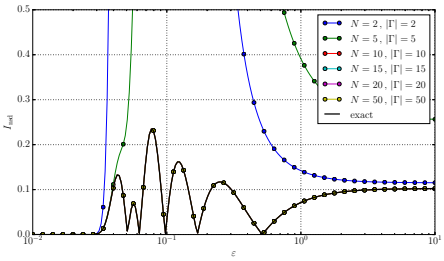


(f) Relative error of the steepest descent method compared to the exact solution.

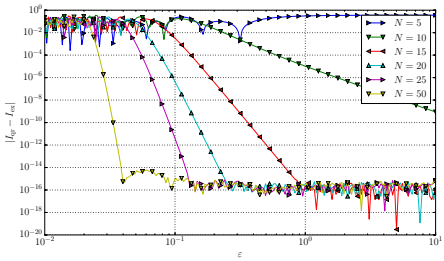
Figure 25: Experiment with ϕ_2 and ϕ_1'. The parameters are: $q = 1.0$, $p = 0.2$, $Q = 0.5$, $P = 2.0i$ and $q' = 1.0$, $p' = -0.2$, $Q' = 2.0$, $P' = 0.5i$.



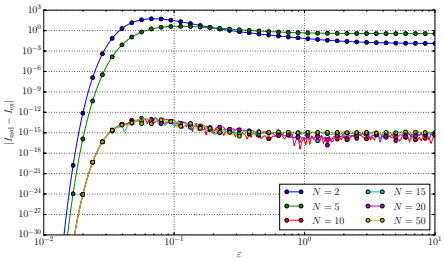
(a) Direct Gauss-Hermite quadrature of size N with a total of $|\Gamma|$ nodes.



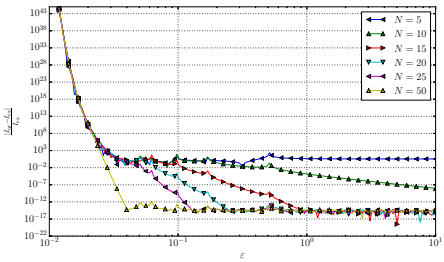
(b) Steepest descent transformation and Gauss-Hermite quadrature of size N with a total of $|\Gamma|$ nodes.



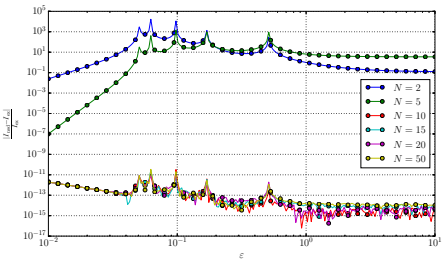
(c) Absolute error of the direct quadrature method compared to the exact solution.



(d) Absolute error of the steepest descent method compared to the exact solution.

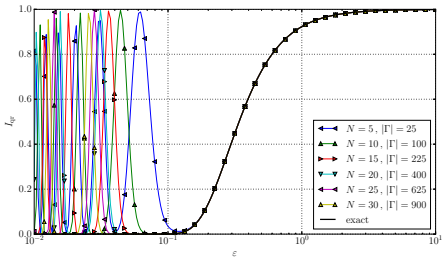


(e) Relative error of the direct quadrature method compared to the exact solution.

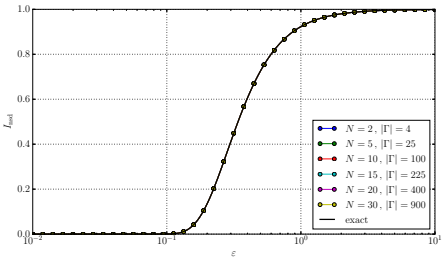


(f) Relative error of the steepest descent method compared to the exact solution.

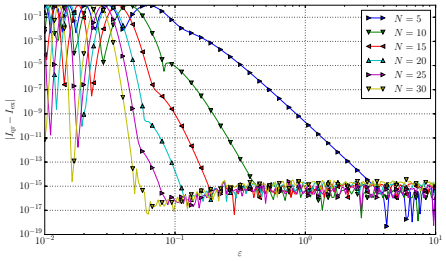
Figure 26: Experiment with ϕ_8 and ϕ_8'. The parameters are: $q = 1.0$, $p = 0.2$, $Q = 0.5$, $P = 2.0\imath$ and $q' = 1.0$, $p' = -0.2$, $Q' = 2.0$, $P' = 0.5\imath$.



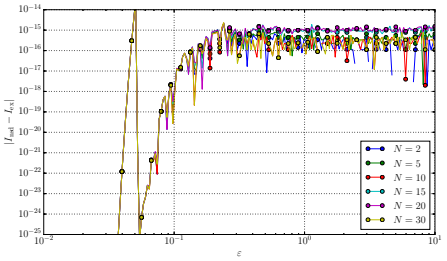
(a) Direct tensor product Gauss-Hermite quadrature of linear size N with a total of $|\Gamma|$ nodes.



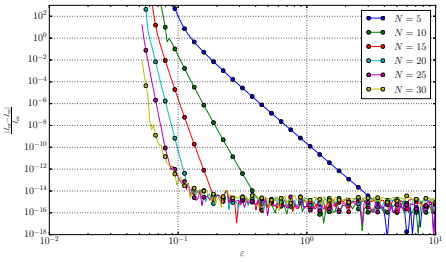
(b) Steepest descent transformation and tensor product Gauss-Hermite quadrature of linear size N with a total of $|\Gamma|$ nodes.



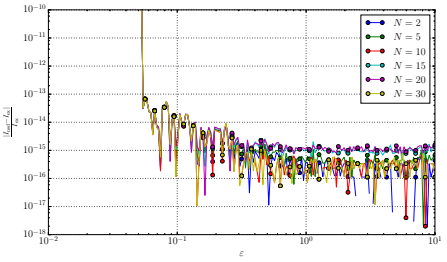
(c) Absolute error of the direct quadrature method compared to the exact solution.



(d) Absolute error of the steepest descent method compared to the exact solution.

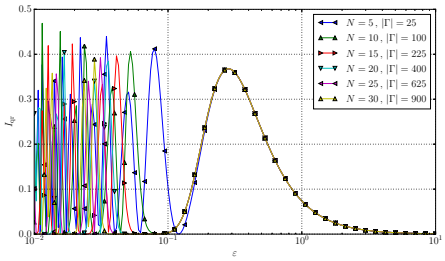


(e) Relative error of the direct quadrature method compared to the exact solution.

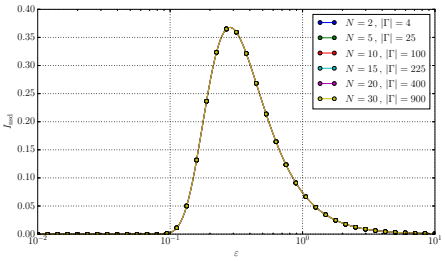


(f) Relative error of the steepest descent method compared to the exact solution.

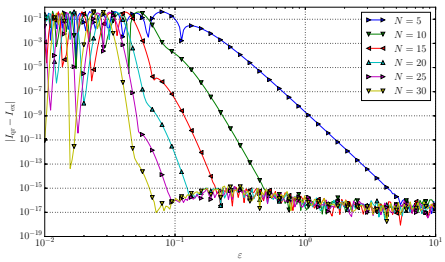
Figure 27: Experiment with $\phi_{0,0}$ and $\phi'_{0,0}$. The parameters are: $\underline{q} = (1, 1)$, $\underline{p} = (-0.2, -0.2)$, $\mathbf{Q} = \mathbf{1}$, $\mathbf{P} = i\mathbf{1}$ and $\underline{q}' = (1, 1)$, $\underline{p}' = (0.2, 0.2)$, $\mathbf{Q}' = \mathbf{1}$, $\mathbf{P}' = i\mathbf{1}$.



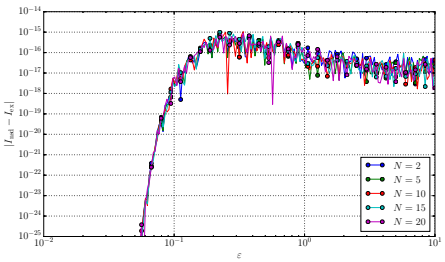
(a) Direct tensor product Gauss-Hermite quadrature of linear size N with a total of $|\Gamma|$ nodes.



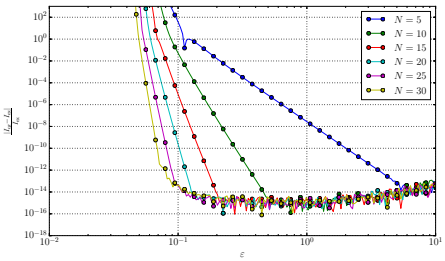
(b) Steepest descent transformation and tensor product Gauss-Hermite quadrature of linear size N with a total of $|\Gamma|$ nodes.



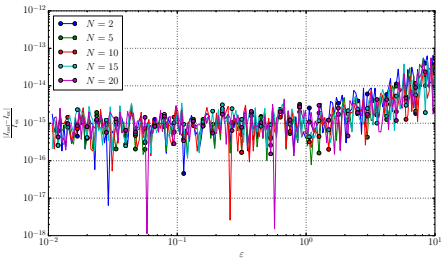
(c) Absolute error of the direct quadrature method compared to the exact solution.



(d) Absolute error of the steepest descent method compared to the exact solution.

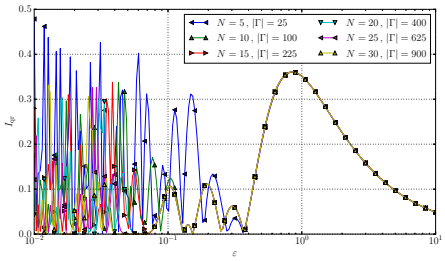


(e) Relative error of the direct quadrature method compared to the exact solution.

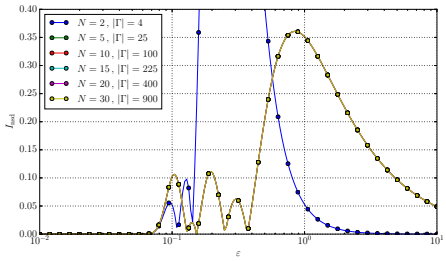


(f) Relative error of the steepest descent method compared to the exact solution.

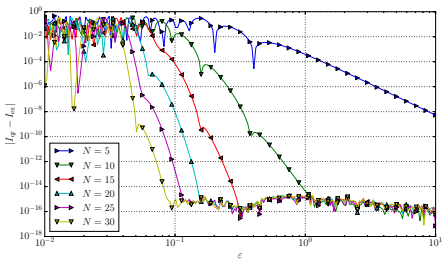
Figure 28: Experiment with $\phi_{0,1}$ and $\phi'_{1,0}$. The parameters are: $\underline{q} = (1, 1)$, $\underline{p} = (-0.2, -0.2)$, $\mathbf{Q} = \mathbf{1}$, $\mathbf{P} = i\mathbf{1}$ and $\underline{q}' = (1, 1)$, $\underline{p}' = (0.2, 0.2)$, $\mathbf{Q}' = \mathbf{1}$, $\mathbf{P}' = i\mathbf{1}$.



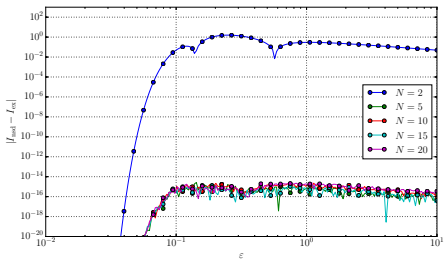
(a) Direct tensor product Gauss-Hermite quadrature of linear size N with a total of $|\Gamma|$ nodes.



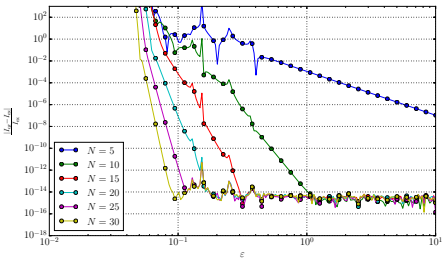
(b) Steepest descent transformation and tensor product Gauss-Hermite quadrature of linear size N with a total of $|\Gamma|$ nodes.



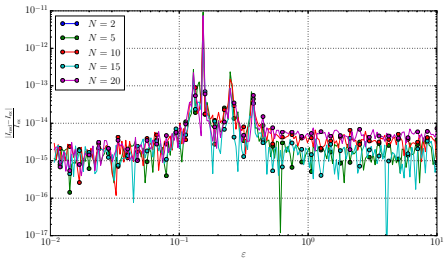
(c) Absolute error of the direct quadrature method compared to the exact solution.



(d) Absolute error of the steepest descent method compared to the exact solution.

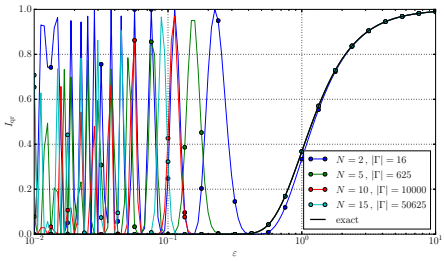


(e) Relative error of the direct quadrature method compared to the exact solution.

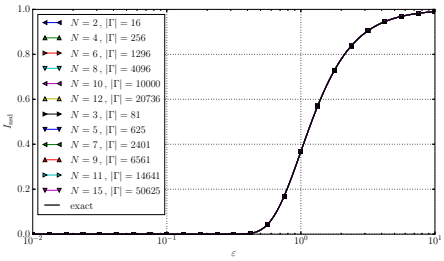


(f) Relative error of the steepest descent method compared to the exact solution.

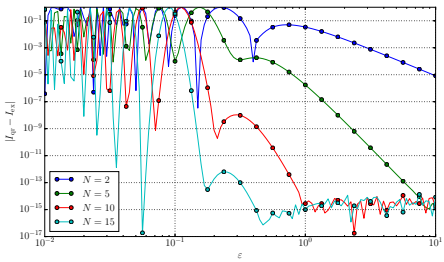
Figure 29: Experiment with $\phi_{2,3}$ and $\phi'_{2,2}$. The parameters are: $\underline{q} = (1, 1)$, $\underline{p} = (-0.2, -0.2)$, $\mathbf{Q} = \mathbf{1}$, $\mathbf{P} = \imath\mathbf{1}$ and $\underline{q}' = (1, 1)$, $\underline{p}' = (0.2, 0.2)$, $\mathbf{Q}' = \mathbf{1}$, $\mathbf{P}' = \imath\mathbf{1}$.



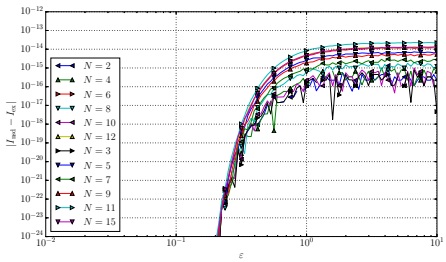
(a) Direct tensor product Gauss-Hermite quadrature of linear size N with a total of $|\Gamma|$ nodes.



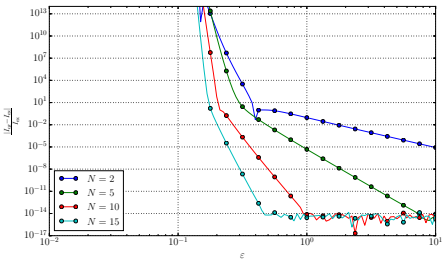
(b) Steepest descent transformation and tensor product Gauss-Hermite quadrature of linear size N with a total of $|\Gamma|$ nodes.



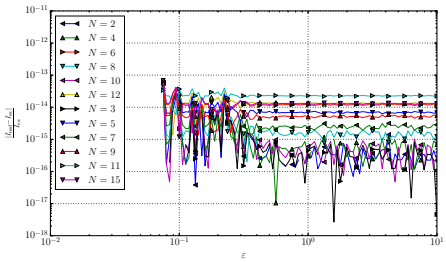
(c) Absolute error of the direct quadrature method compared to the exact solution.



(d) Absolute error of the steepest descent method compared to the exact solution.

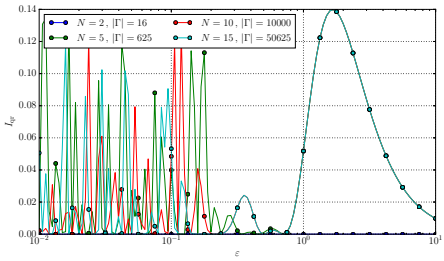


(e) Relative error of the direct quadrature method compared to the exact solution.

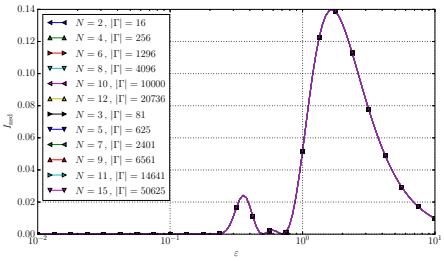


(f) Relative error of the steepest descent method compared to the exact solution.

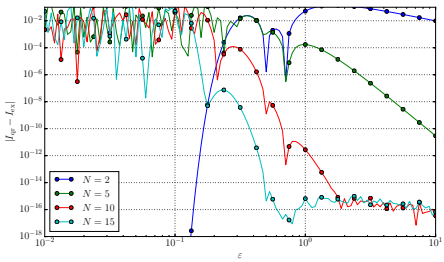
Figure 30: Experiment with $\phi_{0,0,0,0}$ and $\phi'_{0,0,0,0}$. The parameters are: $\underline{q} = (1,1,1,1)$, $\underline{p} = (-0.5,-0.5,-0.5,-0.5)$, $\mathbf{Q} = \mathbf{1}$, $\mathbf{P} = \imath\mathbf{1}$ and $\underline{q}' = (1,1,1,1)$, $\underline{p}' = (0.5,0.5,0.5,0.5)$, $\mathbf{Q}' = \mathbf{1}$, $\mathbf{P}' = \imath\mathbf{1}$.



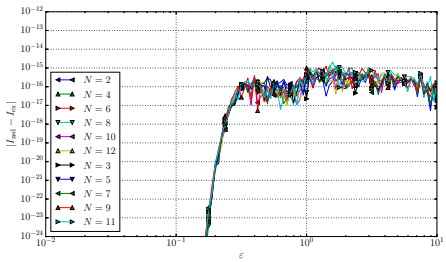
(a) Direct tensor product Gauss-Hermite quadrature of linear size N with a total of $|\Gamma|$ nodes.



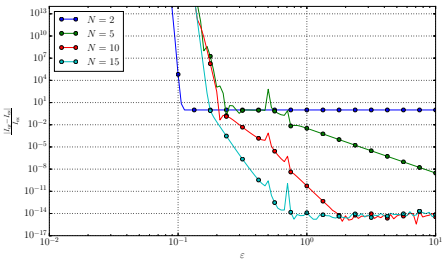
(b) Steepest descent transformation and tensor product Gauss-Hermite quadrature of linear size N with a total of $|\Gamma|$ nodes.



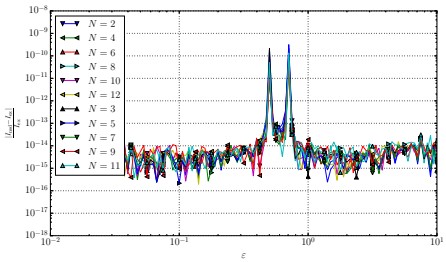
(c) Absolute error of the direct quadrature method compared to the exact solution.



(d) Absolute error of the steepest descent method compared to the exact solution.



(e) Relative error of the direct quadrature method compared to the exact solution.



(f) Relative error of the steepest descent method compared to the exact solution.

Figure 31: Experiment with $\phi_{2,1,2,1}$ and $\phi'_{1,1,1,1}$. The parameters are: $\underline{q} = (1,1,1,1)$, $\underline{p} = (-0.5,-0.5,-0.5,-0.5)$, $\mathbf{Q} = \mathbf{1}$, $\mathbf{P} = \imath\mathbf{1}$ and $\underline{q}' = (1,1,1,1)$, $\underline{p}' = (0.5,0.5,0.5,0.5)$, $\mathbf{Q}' = \mathbf{1}$, $\mathbf{P}' = \imath\mathbf{1}$.

8.3 Going to higher dimensions

As seen in the last section, after the steepest descent transformation we still need a quadrature albeit with much less quadrature points. In the case of higher dimensionality (at least $D > 2$) of the original problem this full tensor product in general is very expensive. Therefore we seek for a replacement that is cheaper to compute but still accurate. The Smolyak approach is an obvious candidate. If it really is well-suited depends also on the multi-indices $\underline{k}$ and $\underline{l}$ of the wavepackets ϕ and ϕ' which have to be sparse enough. But we assume that this is the case for now. Then we end up in a chain as shown in Figure 32, first performing the steepest descent transformation to get rid of the oscillations then using the Smolyak construction to lessen the curse of dimensionality.

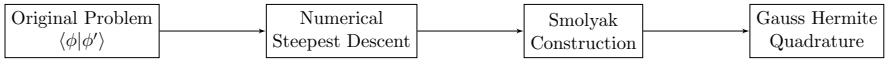


Figure 32: Overview of the computation seen as chain of transformations.

For this to work out we rewrite the Smolyak construction into a less well known version such that in can act as a simple transformation of a set of Gauss-Hermite quadrature nodes connecting to the numerical steepest descent step. The Smolyak construction of level $k \geq 1$ in D dimensions can be written as:

$$S_{D,k} := \sum_{q=k-D}^{k-1} (-1)^{k-1-q} \binom{D-1}{k-1-q} \sum_{\substack{\underline{l} \in \mathbb{N}^D \\ \|\underline{l}\|_1 = D+q}} (Q_{l_1} \otimes \cdots \otimes Q_{l_D})$$

which is known as the combination technique. The formula was first derived in [21]. This version stresses that the Smolyak construction is nothing more than a certain sum of specific smaller tensor products of one-dimensional quadrature rules $\{Q_i\}_{i \in \mathbb{N}}$. Our own implementation is based mainly on the details explained in [14] and [9].
However, this setup as shown will not work too well and, even worse, yield a much *larger* number of nodes compared to the simple full tensor product under some circumstances. The reason is that the Gauss-Hermite quadrature points are not *nested* while this is a central prerequisite for the Smolyak construction to work and reduce complexity.
To resolve this issue, we look for sets of nested points that can be fed into the Smolyak construction. Luckily there exist the so-called Genz-Keister quadrature points which can be build for the case of a Gaussian integral, too. For further details on the Genz-Keister construction, see the original paper [8] as well as our own work [5] and references there in. Using these points instead of the Gauss-Hermite nodes will result in a chain as shown in Figure 33.

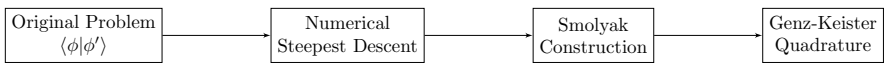


Figure 33: Overview of the computation seen as chain of transformations.

These Genz-Keister quadrature nodes can be constructed not only for the one-dimensional case but in any number of dimensions with moderate effort. Even better, one can prove [18] that the resulting multi-dimensional Genz-Keister construction is equivalent to the Smolyak construction starting from one-dimensional Genz-Keister nodes. Since there are much less Genz-Keister nodes compared to the full tensor of Gauss-Hermite nodes, as shown in Figure 34, we can largely reduce the computational effort. Note however that for $D \leq 2$ they are unprofitable because they encompass more nodes that a simple Gauss-Hermite tensor product.

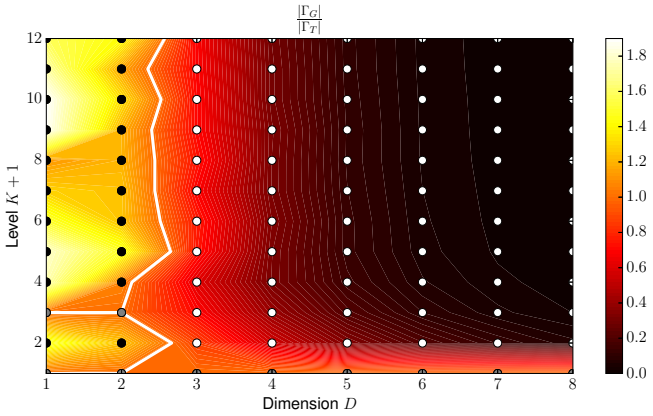


Figure 34: Heat map of the ratio $\frac{|\Gamma_G|}{|\Gamma_T|}$ of number $|\Gamma_G|$ of Genz-Keister points and number $|\Gamma_T|$ of Gauss-Hermite tensor product points for dimensions D up to 8 and level $K \leq 12$. White dots are D, K combinations where Genz-Keister is advantageous, while for black dots Genz-Keister is worse and for gray dots the ratio equals 1.

Inside the algorithm producing these nodes we can at the same time also explicitly compute the necessary matching weights for quadrature. At the end of the day we obtain the relatively simple chain shown in 35 where we plug the set of multi-dimensional Genz-Keister nodes $\gamma_i \in \mathbb{R}^D$ directly into the steepest descent transformation. However, we should keep in mind that this reduction is not always applicable or profitable, depending on $\underline{k}$ and $\underline{l}$.

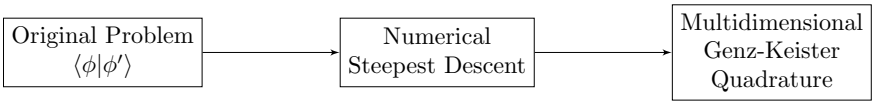
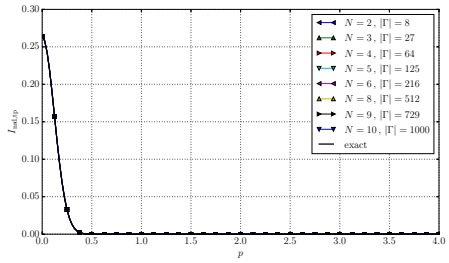


Figure 35: Overview of the computation seen as chain of transformations.

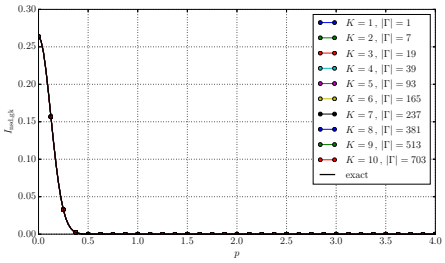
In the remainder of this section we will perform numerical experiments and compare this approach based on Genz-Keister quadrature rules to the full tensor ansatz using Gauss-Hermite node weight pairs. Again we study the convergence of the quadrature error in the same setup as in the last section. First we measure the error for oscillations caused by increasing momentum. The second setup

fixes momentum and varies the semiclassical scaling parameter ε, also resulting in oscillatory integrands.

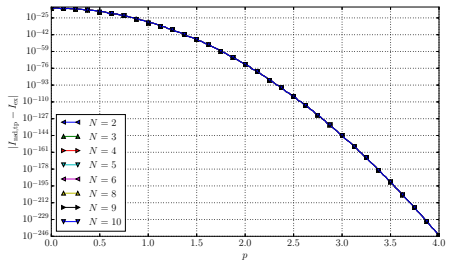
8.3.1 Convergence in $|p|$



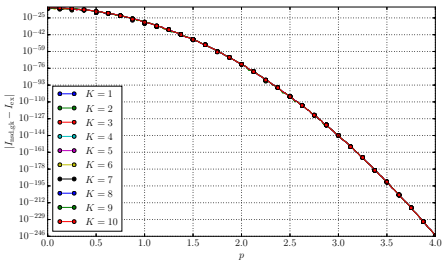
(a) Tensor product of linear size N with a total of $|\Gamma|$ quadrature nodes.



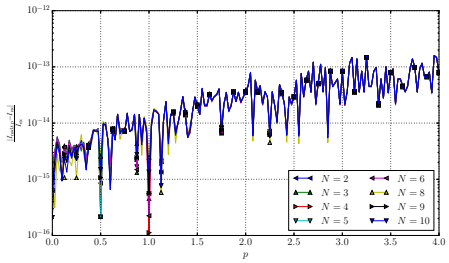
(b) Smolyak construction of level K with a total of $|\Gamma|$ quadrature nodes.



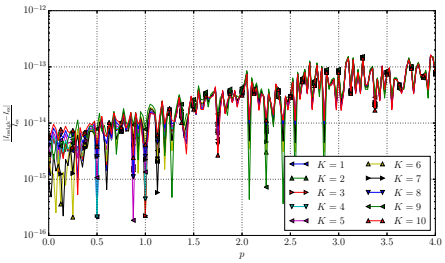
(c) Absolute error of the tensor product ansatz compared to the exact solution.



(d) Absolute error of the Smolyak construction compared to the exact solution.

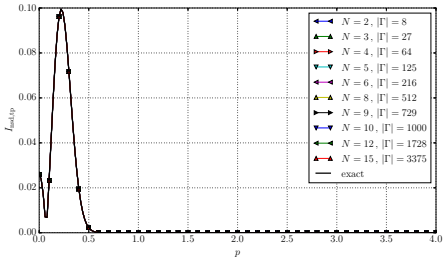


(e) Relative error of the tensor product ansatz compared to the exact solution.

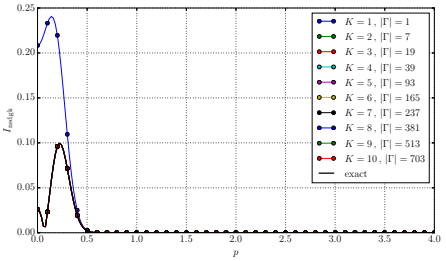


(f) Relative error of the Smolyak construction compared to the exact solution.

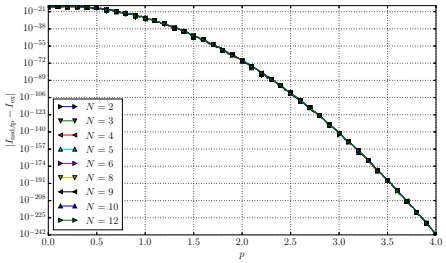
Figure 36: Experiment with $\phi_{0,0,0}$ and $\phi'_{0,0,0}$ in 3 dimensions. The parameters are: $\underline{q} = (1.2, 1.2, 1.2)$, $\underline{p} = (1, 1, 1)$, $\mathbf{Q} = \mathbf{1}$, $\mathbf{P} = \imath\mathbf{1}$ and $\underline{q}' = (0.8, 0.8, 0.8)$, $\underline{p}' = (-1, -1, -1)$, $\mathbf{Q}' = \mathbf{1}$, $\mathbf{P}' = \imath\mathbf{1}$ with $\varepsilon = 0.3$.



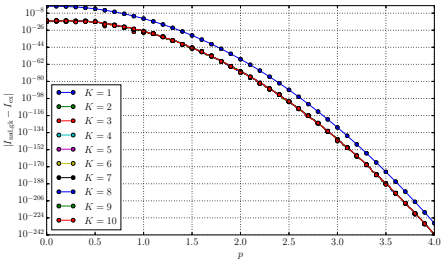
(a) Tensor product of linear size N with a total of $|\Gamma|$ quadrature nodes.



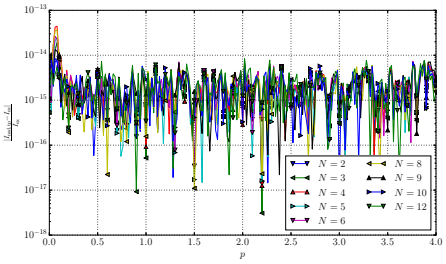
(b) Smolyak construction of level K with a total of $|\Gamma|$ quadrature nodes.



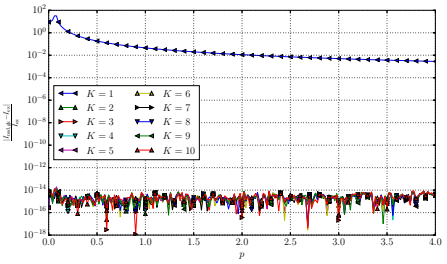
(c) Absolute error of the tensor product ansatz compared to the exact solution.



(d) Absolute error of the Smolyak construction compared to the exact solution.

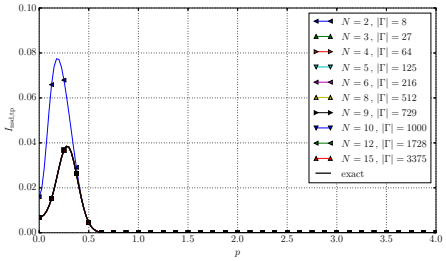


(e) Relative error of the tensor product ansatz compared to the exact solution.

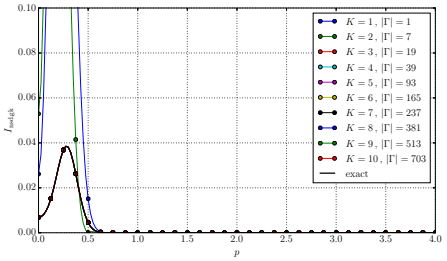


(f) Relative error of the Smolyak construction compared to the exact solution.

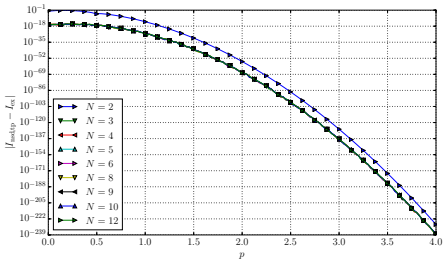
Figure 37: Experiment with $\phi_{0,1,1}$ and $\phi'_{1,1,0}$ in 3 dimensions. The parameters are: $\underline{q} = 1.2(1,\dots,1)$, $\underline{p} = (1,\dots,1)$, $\mathbf{Q} = \mathbf{1}$, $\mathbf{P} = \imath\mathbf{1}$ and $\underline{q}' = 0.8(1,\dots,1)$, $\underline{p}' = -(1,\dots,1)$, $\mathbf{Q}' = \mathbf{1}$, $\mathbf{P}' = \imath\mathbf{1}$ with $\varepsilon = 0.3$.



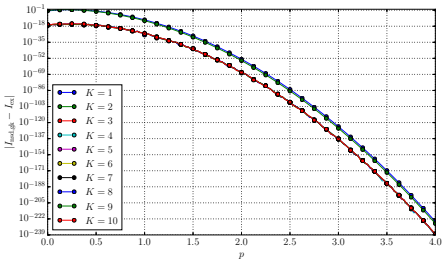
(a) Tensor product of linear size N with a total of $|\Gamma|$ quadrature nodes.



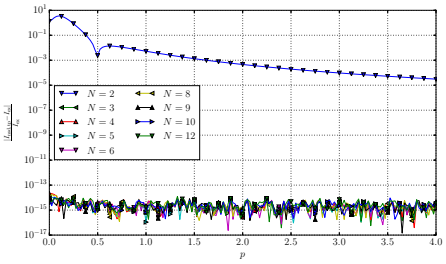
(b) Smolyak construction of level K with a total of $|\Gamma|$ quadrature nodes.



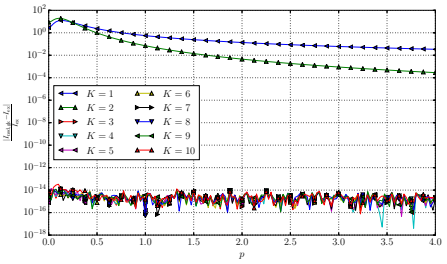
(c) Absolute error of the tensor product ansatz compared to the exact solution.



(d) Absolute error of the Smolyak construction compared to the exact solution.

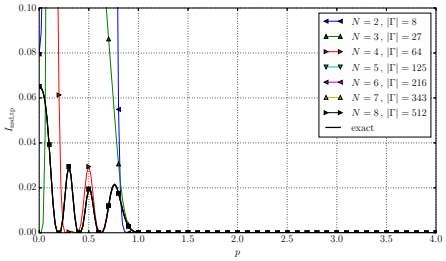


(e) Relative error of the tensor product ansatz compared to the exact solution.

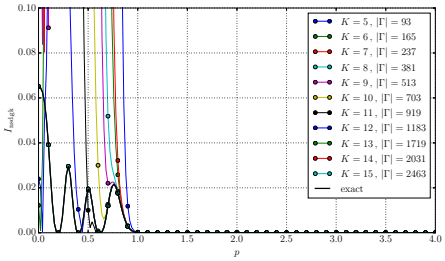


(f) Relative error of the Smolyak construction compared to the exact solution.

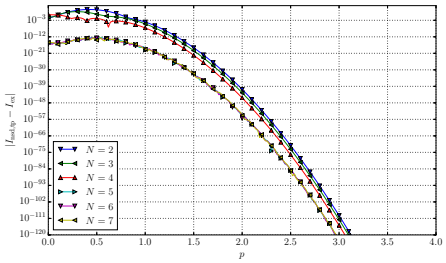
Figure 38: Experiment with $\phi_{0,4,0}$ and $\phi'_{4,0,0}$ in 3 dimensions. The parameters are: $\underline{q} = 1.2(1,\dots,1)$, $\underline{p} = (1,\dots,1)$, $\mathbf{Q} = \mathbf{1}$, $\mathbf{P} = \imath\mathbf{1}$ and $\underline{q}' = 0.8(1,\dots,1)$, $\underline{p}' = -(1,\dots,1)$, $\mathbf{Q}' = \mathbf{1}$, $\mathbf{P}' = \imath\mathbf{1}$ with $\varepsilon = 0.3$.



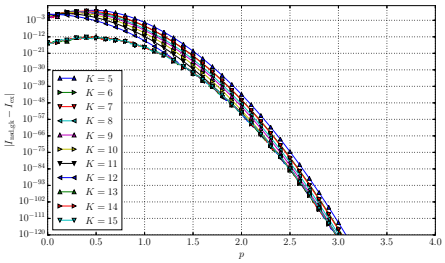
(a) Tensor product of linear size N with a total of $|\Gamma|$ quadrature nodes.



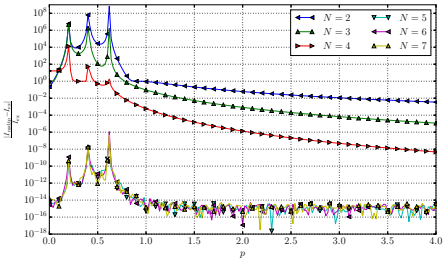
(b) Smolyak construction of level K with a total of $|\Gamma|$ quadrature nodes.



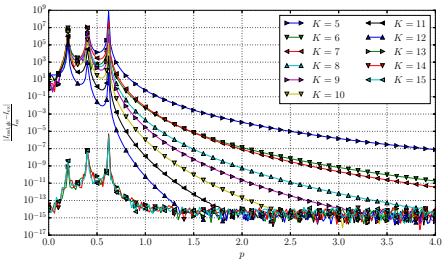
(c) Absolute error of the tensor product ansatz compared to the exact solution.



(d) Absolute error of the Smolyak construction compared to the exact solution.

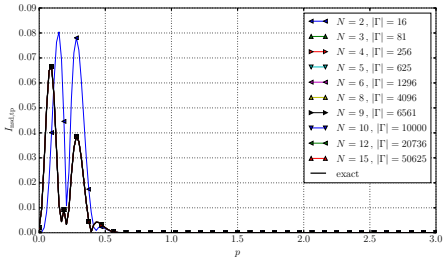


(e) Relative error of the tensor product ansatz compared to the exact solution.

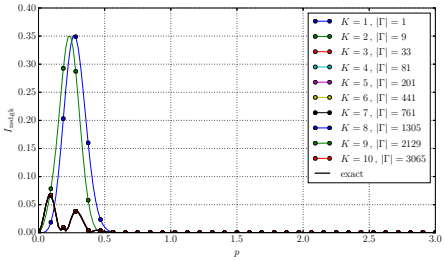


(f) Relative error of the Smolyak construction compared to the exact solution.

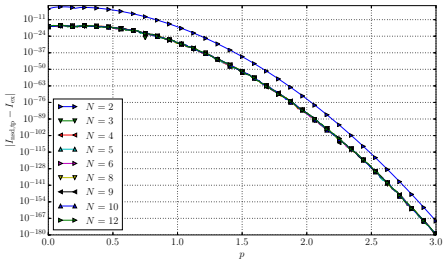
Figure 39: Experiment with $\phi_{4,4,4}$ and $\phi'_{4,4,4}$ in 3 dimensions. The parameters are: $\underline{q} = 1.2(1,\dots,1)$, $\underline{p} = (1,\dots,1)$, $\mathbf{Q} = \mathbf{1}$, $\mathbf{P} = \imath\mathbf{1}$ and $\underline{q}' = 0.8(1,\dots,1)$, $\underline{p}' = -(1,\dots,1)$, $\mathbf{Q}' = \mathbf{1}$, $\mathbf{P}' = \imath\mathbf{1}$ with $\varepsilon = 0.3$.



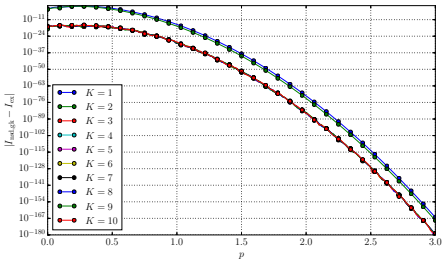
(a) Tensor product of linear size N with a total of $|\Gamma|$ quadrature nodes.



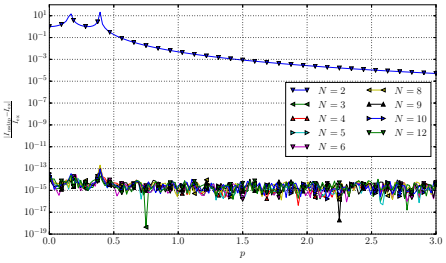
(b) Smolyak construction of level K with a total of $|\Gamma|$ quadrature nodes.



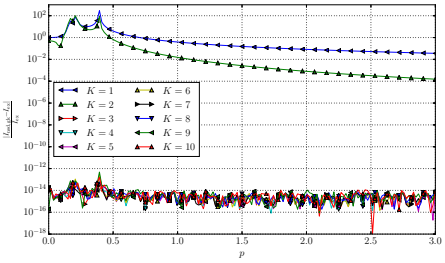
(c) Absolute error of the tensor product ansatz compared to the exact solution.



(d) Absolute error of the Smolyak construction compared to the exact solution.

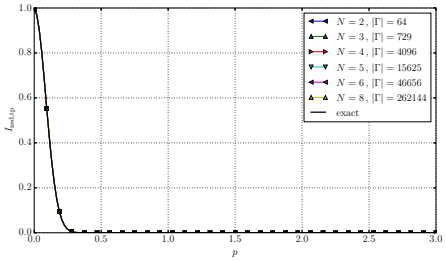


(e) Relative error of the tensor product ansatz compared to the exact solution.

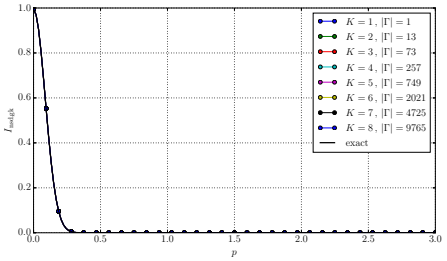


(f) Relative error of the Smolyak construction compared to the exact solution.

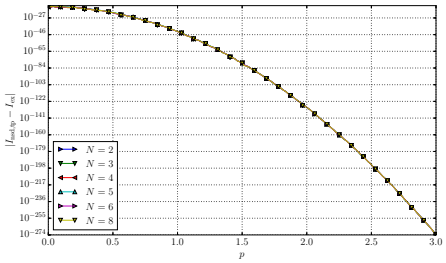
Figure 40: Experiment with $\phi_{0,2,1,1}$ and $\phi'_{1,2,1,0}$ in 4 dimensions. The parameters are: $\underline{q} = 1.01(1,\dots,1)$, $\underline{p} = (1,\dots,1)$, $\mathbf{Q} = \mathbf{1}$, $\mathbf{P} = \imath\mathbf{1}$ and $\underline{q}' = 0.99(1,\dots,1)$, $\underline{p}' = -(1,\dots,1)$, $\mathbf{Q}' = \mathbf{1}$, $\mathbf{P}' = \imath\mathbf{1}$ with $\varepsilon = 0.3$.



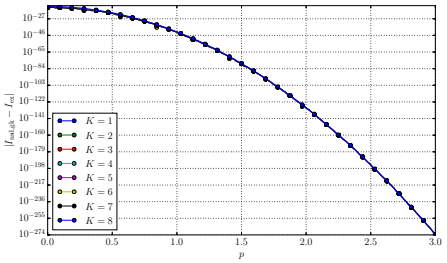
(a) Tensor product of linear size N with a total of $|\Gamma|$ quadrature nodes.



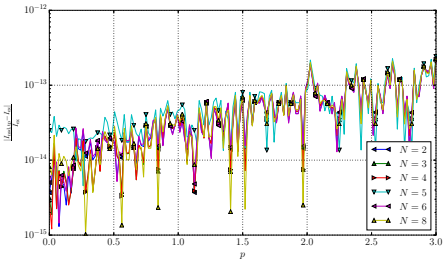
(b) Smolyak construction of level K with a total of $|\Gamma|$ quadrature nodes.



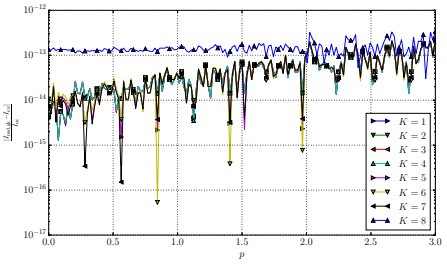
(c) Absolute error of the tensor product ansatz compared to the exact solution.



(d) Absolute error of the Smolyak construction compared to the exact solution.

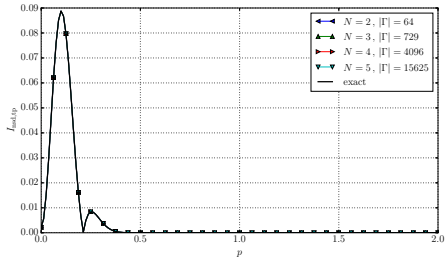


(e) Relative error of the tensor product ansatz compared to the exact solution.

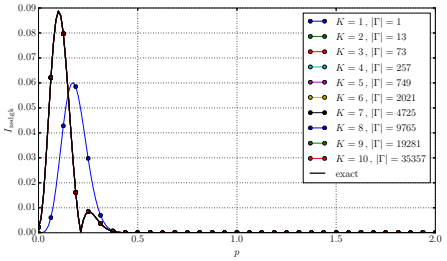


(f) Relative error of the Smolyak construction compared to the exact solution.

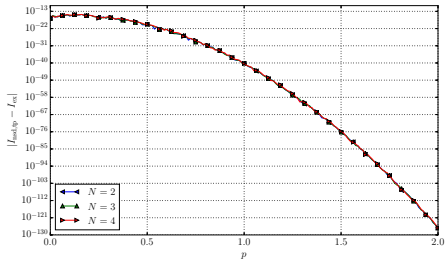
Figure 41: Experiment with $\phi_{\underline{0}}$ and $\phi'_{\underline{0}}$ in 6 dimensions. The parameters are: $\underline{q} = 1.01(1, \ldots, 1)$, $\underline{p} = (1, \ldots, 1)$, $\mathbf{Q} = \mathbf{1}$, $\mathbf{P} = \imath\mathbf{1}$ and $\underline{q}' = 0.99(1, \ldots, 1)$, $\underline{p}' = -(1, \ldots, 1)$, $\mathbf{Q}' = \mathbf{1}$, $\mathbf{P}' = \imath\mathbf{1}$ with $\varepsilon = 0.3$.



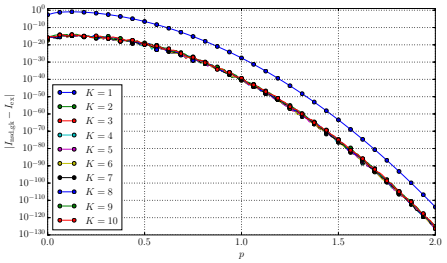
(a) Tensor product of linear size N with a total of $|\Gamma|$ quadrature nodes.



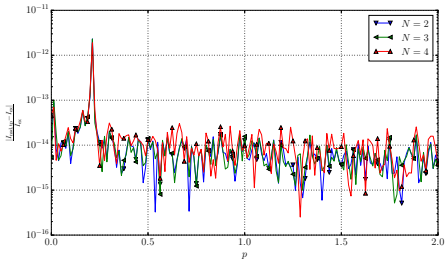
(b) Smolyak construction of level K with a total of $|\Gamma|$ quadrature nodes.



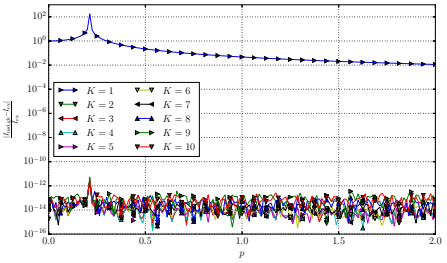
(c) Absolute error of the tensor product ansatz compared to the exact solution.



(d) Absolute error of the Smolyak construction compared to the exact solution.

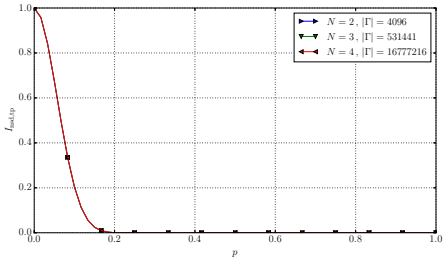


(e) Relative error of the tensor product ansatz compared to the exact solution.

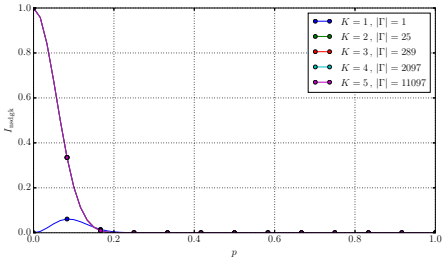


(f) Relative error of the Smolyak construction compared to the exact solution.

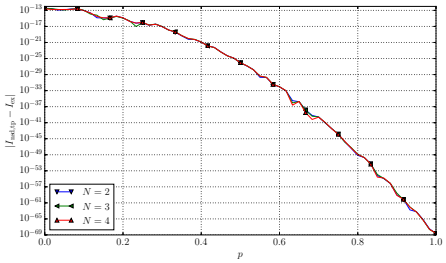
Figure 42: Experiment with $\phi_{1,0,1,0,0,0}$ and $\phi'_{1,1,0,0,0,0}$ in 6 dimensions. The parameters are: $\underline{q} = 1.01(1,\ldots,1)$, $\underline{p} = (1,\ldots,1)$, $\mathbf{Q} = \mathbf{1}$, $\mathbf{P} = \imath\mathbf{1}$ and $\underline{q}' = 0.99(1,\ldots,1)$, $\underline{p}' = -(1,\ldots,1)$, $\mathbf{Q}' = \mathbf{1}$, $\mathbf{P}' = \imath\mathbf{1}$ with $\varepsilon = 0.3$.



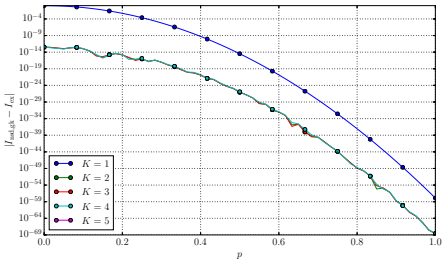
(a) Tensor product of linear size N with a total of $|\Gamma|$ quadrature nodes.



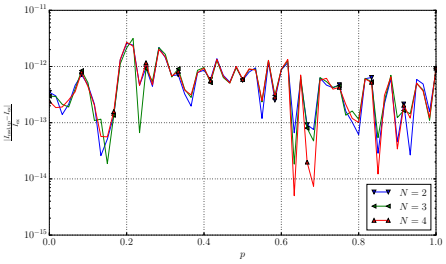
(b) Smolyak construction of level K with a total of $|\Gamma|$ quadrature nodes.



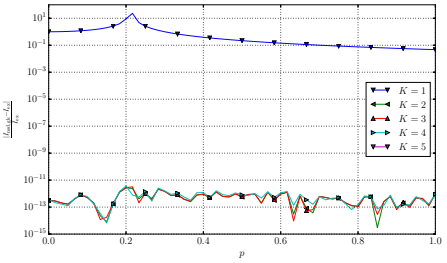
(c) Absolute error of the tensor product ansatz compared to the exact solution.



(d) Absolute error of the Smolyak construction compared to the exact solution.



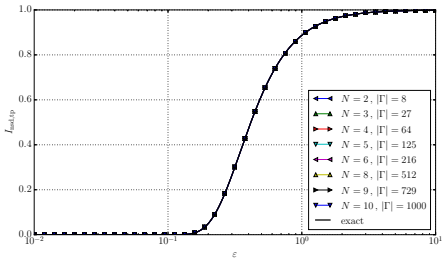
(e) Relative error of the tensor product ansatz compared to the exact solution.



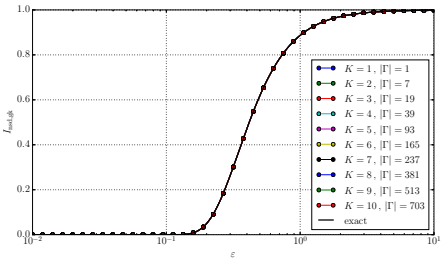
(f) Relative error of the Smolyak construction compared to the exact solution.

Figure 43: Experiment with $\phi_{1,0,\dots,0}$ and $\phi'_{1,0,\dots,0}$ in 12 dimensions. The parameters are: $\underline{q} = 1.0001(1,\dots,1)$, $\underline{p} = (1,\dots,1)$, $\mathbf{Q} = \mathbf{1}$, $\mathbf{P} = \imath\mathbf{1}$ and $\underline{q}' = 0.9999(1,\dots,1)$, $\underline{p}' = -(1,\dots,1)$, $\mathbf{Q}' = \mathbf{1}$, $\mathbf{P}' = \imath\mathbf{1}$ with $\varepsilon = 0.3$.

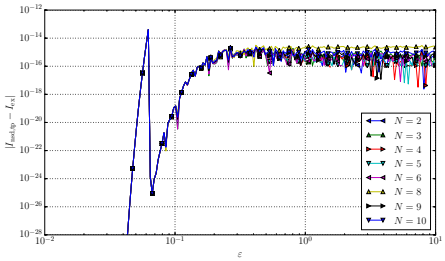
8.3.2 Convergence in ε



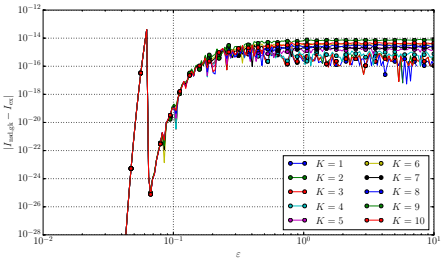
(a) Tensor product of linear size N with a total of $|\Gamma|$ quadrature nodes.



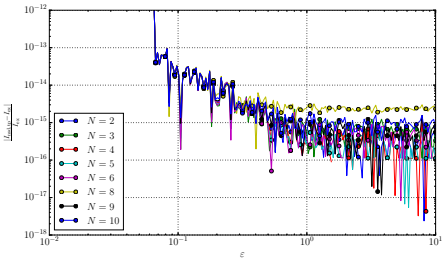
(b) Smolyak construction of level K with a total of $|\Gamma|$ quadrature nodes.



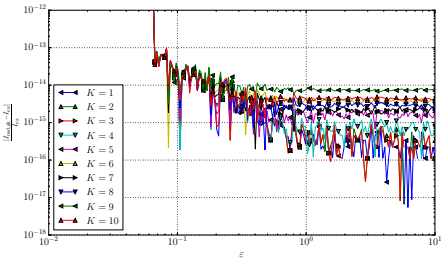
(c) Absolute error of the tensor product ansatz compared to the exact solution.



(d) Absolute error of the Smolyak construction compared to the exact solution.

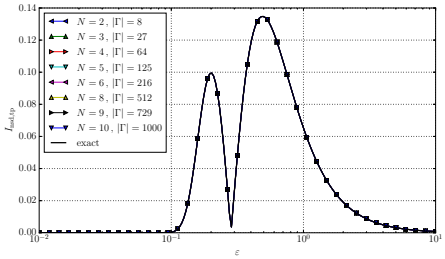


(e) Relative error of the tensor product ansatz compared to the exact solution.

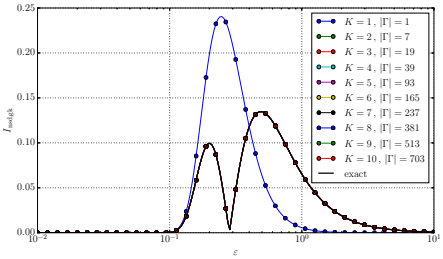


(f) Relative error of the Smolyak construction compared to the exact solution.

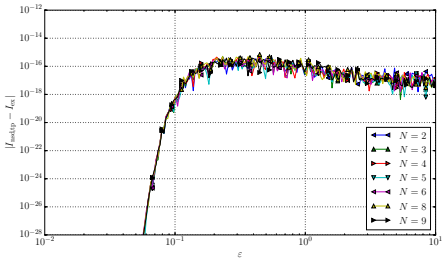
Figure 44: Experiment with $\phi_{0,0,0}$ and $\phi'_{0,0,0}$ in 3 dimensions. The parameters are: $\underline{q} = (1,1,1)$, $\underline{p} = (-0.2,-0.2,-0.2)$, $\mathbf{Q} = \mathbf{1}$, $\mathbf{P} = \imath\mathbf{1}$ and $\underline{q}' = (1,1,1)$, $\underline{p}' = (0.2,0.2,0.2)$, $\mathbf{Q}' = \mathbf{1}$, $\mathbf{P}' = \imath\mathbf{1}$.



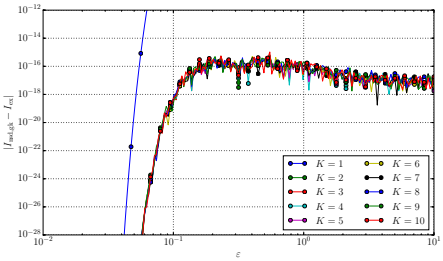
(a) Tensor product of linear size N with a total of $|\Gamma|$ quadrature nodes.



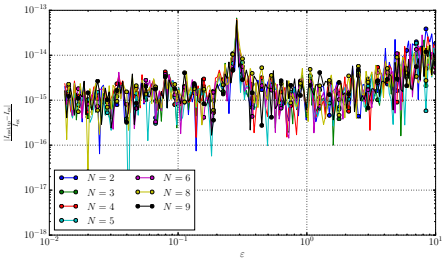
(b) Smolyak construction of level K with a total of $|\Gamma|$ quadrature nodes.



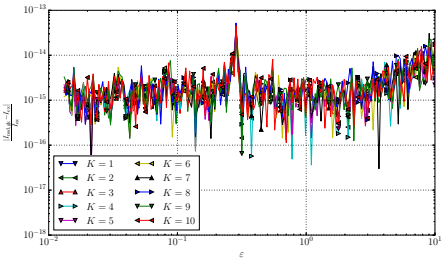
(c) Absolute error of the tensor product ansatz compared to the exact solution.



(d) Absolute error of the Smolyak construction compared to the exact solution.

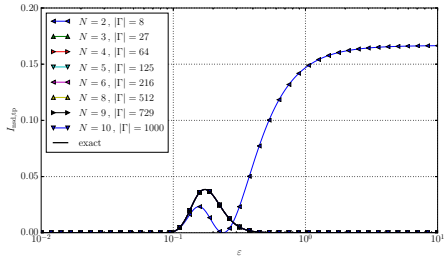


(e) Relative error of the tensor product ansatz compared to the exact solution.

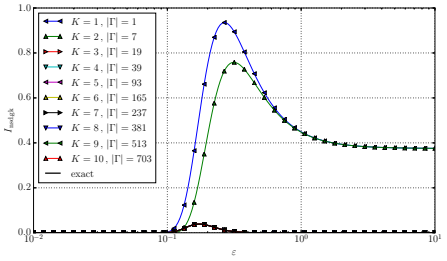


(f) Relative error of the Smolyak construction compared to the exact solution.

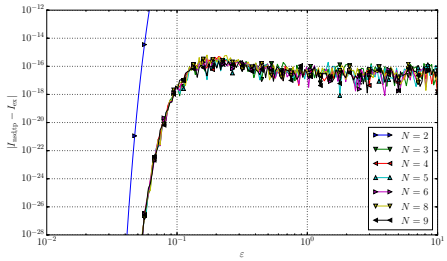
Figure 45: Experiment with $\phi_{1,1,0}$ and $\phi'_{0,1,1}$ in 3 dimensions. The parameters are: $\underline{q} = (1,1,1)$, $\underline{p} = (-0.2,-0.2,-0.2)$, $\mathbf{Q} = \mathbf{1}$, $\mathbf{P} = \imath\mathbf{1}$ and $\underline{q}' = (1,1,1)$, $\underline{p}' = (0.2,0.2,0.2)$, $\mathbf{Q}' = \mathbf{1}$, $\mathbf{P}' = \imath\mathbf{1}$.



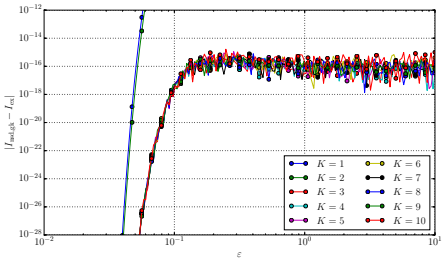
(a) Tensor product of linear size N with a total of $|\Gamma|$ quadrature nodes.



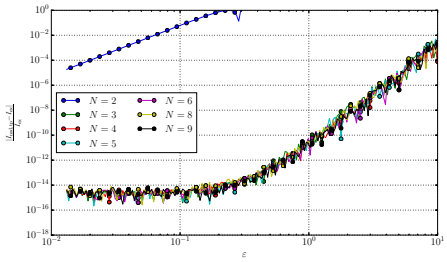
(b) Smolyak construction of level K with a total of $|\Gamma|$ quadrature nodes.



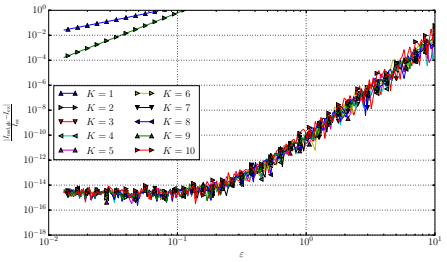
(c) Absolute error of the tensor product ansatz compared to the exact solution.



(d) Absolute error of the Smolyak construction compared to the exact solution.

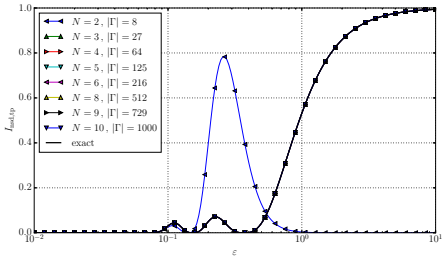


(e) Relative error of the tensor product ansatz compared to the exact solution.

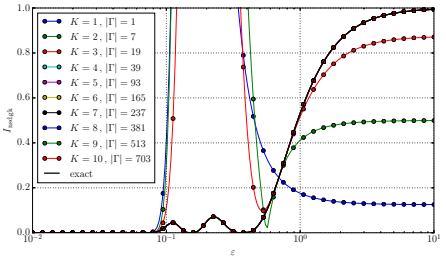


(f) Relative error of the Smolyak construction compared to the exact solution.

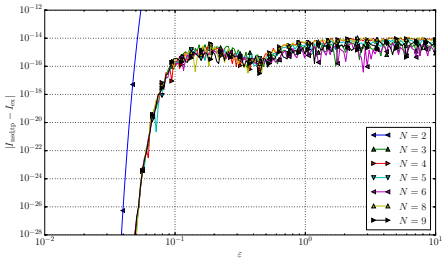
Figure 46: Experiment with $\phi_{4,0,0}$ and $\phi'_{0,4,0}$ in 3 dimensions. The parameters are: $\underline{q} = (1,1,1)$, $\underline{p} = (-0.2,-0.2,-0.2)$, $\mathbf{Q} = \mathbf{1}$, $\mathbf{P} = \imath\mathbf{1}$ and $\underline{q}' = (1,1,1)$, $\underline{p}' = (0.2,0.2,0.2)$, $\mathbf{Q}' = \mathbf{1}$, $\mathbf{P}' = \imath\mathbf{1}$.



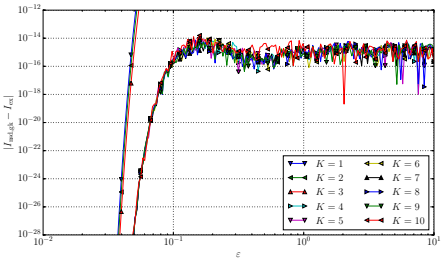
(a) Tensor product of linear size N with a total of $|\Gamma|$ quadrature nodes.



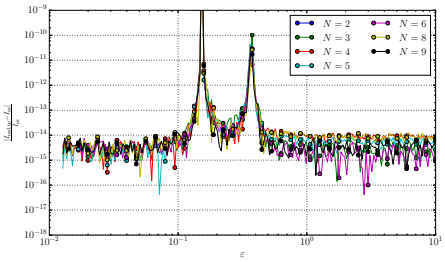
(b) Smolyak construction of level K with a total of $|\Gamma|$ quadrature nodes.



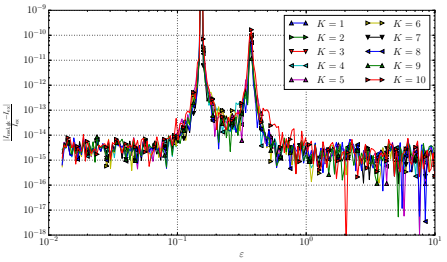
(c) Absolute error of the tensor product ansatz compared to the exact solution.



(d) Absolute error of the Smolyak construction compared to the exact solution.

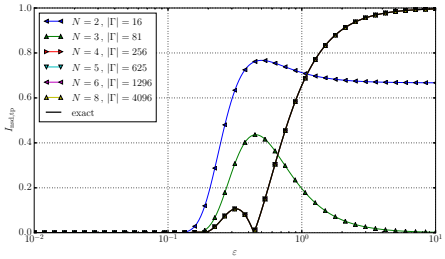


(e) Relative error of the tensor product ansatz compared to the exact solution.

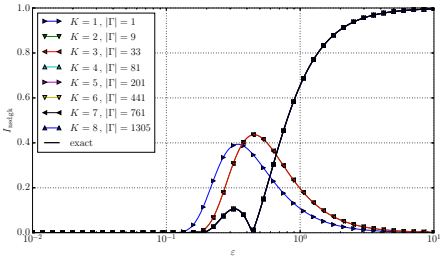


(f) Relative error of the Smolyak construction compared to the exact solution.

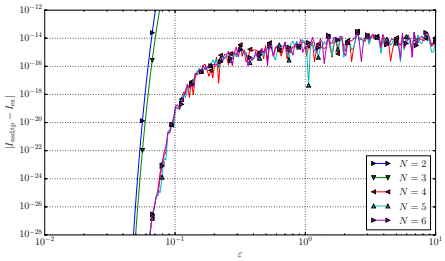
Figure 47: Experiment with $\phi_{2,2,2}$ and $\phi'_{2,2,2}$ in 3 dimensions. The parameters are: $\underline{q} = (1,1,1)$, $\underline{p} = (-0.2,-0.2,-0.2)$, $\mathbf{Q} = \mathbf{1}$, $\mathbf{P} = \imath\mathbf{1}$ and $\underline{q}' = (1,1,1)$, $\underline{p}' = (0.2,0.2,0.2)$, $\mathbf{Q}' = \mathbf{1}$, $\mathbf{P}' = \imath\mathbf{1}$.



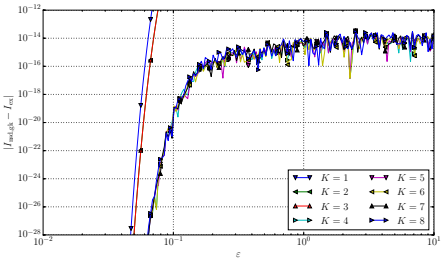
(a) Tensor product of linear size N with a total of $|\Gamma|$ quadrature nodes.



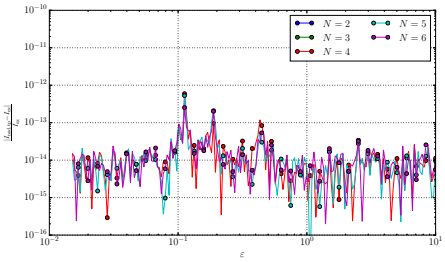
(b) Smolyak construction of level K with a total of $|\Gamma|$ quadrature nodes.



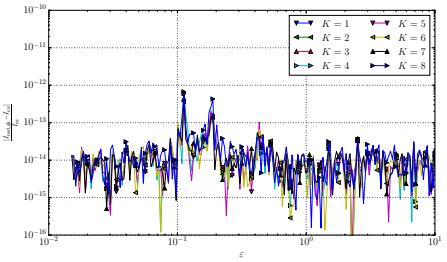
(c) Absolute error of the tensor product ansatz compared to the exact solution.



(d) Absolute error of the Smolyak construction compared to the exact solution.

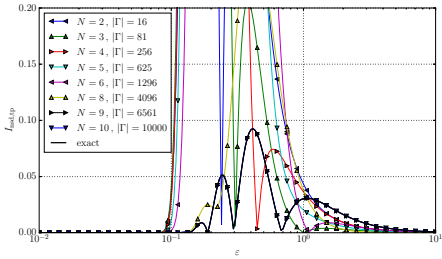


(e) Relative error of the tensor product ansatz compared to the exact solution.

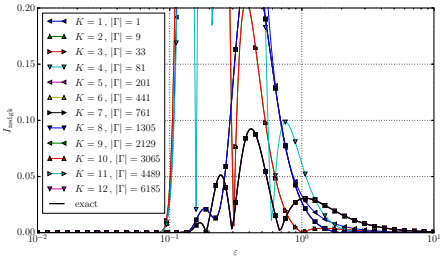


(f) Relative error of the Smolyak construction compared to the exact solution.

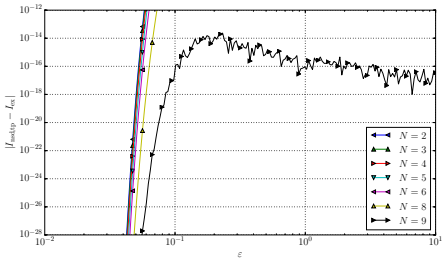
Figure 48: Experiment with $\phi_{3,0,0,0}$ and $\phi'_{3,0,0,0}$ in 4 dimensions. The parameters are: $\underline{q} = (1, 1, 1, 1)$, $\underline{p} = (-0.2, -0.2, -0.2, -0.2)$, $\mathbf{Q} = \mathbf{1}$, $\mathbf{P} = \imath\mathbf{1}$ and $\underline{q}' = (1, 1, 1, 1)$, $\underline{p}' = (0.2, 0.2, 0.2, 0.2)$, $\mathbf{Q}' = \mathbf{1}$, $\mathbf{P}' = \imath\mathbf{1}$.



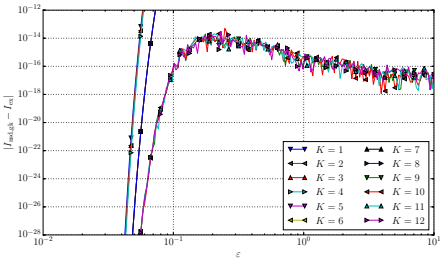
(a) Tensor product of linear size N with a total of $|\Gamma|$ quadrature nodes.



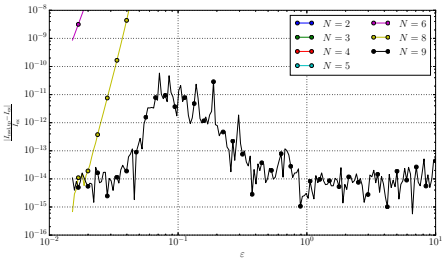
(b) Smolyak construction of level K with a total of $|\Gamma|$ quadrature nodes.



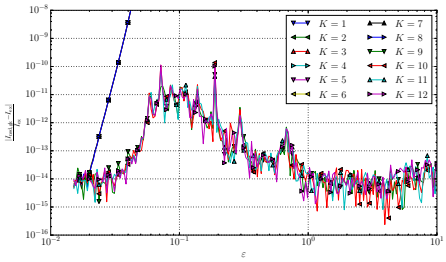
(c) Absolute error of the tensor product ansatz compared to the exact solution.



(d) Absolute error of the Smolyak construction compared to the exact solution.

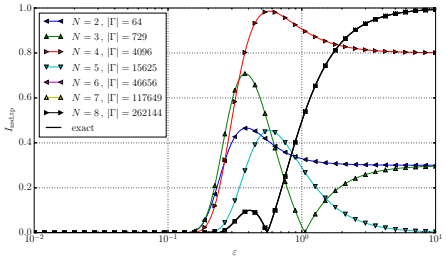


(e) Relative error of the tensor product ansatz compared to the exact solution.

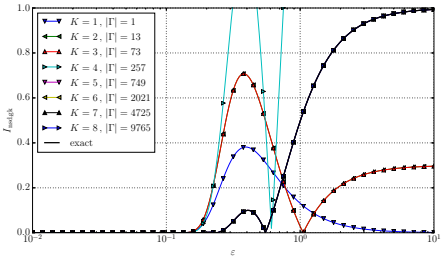


(f) Relative error of the Smolyak construction compared to the exact solution.

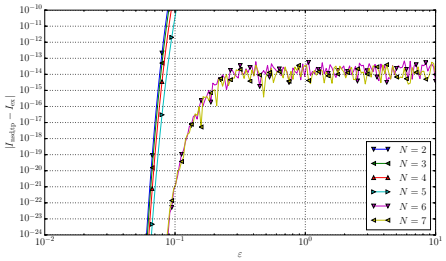
Figure 49: Experiment with $\phi_{8,0,1,0}$ and $\phi'_{8,1,0,0}$ in 4 dimensions. The parameters are: $\underline{q} = (1,1,1,1)$, $\underline{p} = (-0.2,-0.2,-0.2,-0.2)$, $\mathbf{Q} = \mathbf{1}$, $\mathbf{P} = \imath\mathbf{1}$ and $\underline{q}' = (1,1,1,1)$, $\underline{p}' = (0.2,0.2,0.2,0.2)$, $\mathbf{Q}' = \mathbf{1}$, $\mathbf{P}' = \imath\mathbf{1}$.



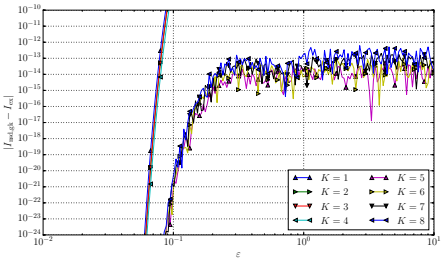
(a) Tensor product of linear size N with a total of $|\Gamma|$ quadrature nodes.



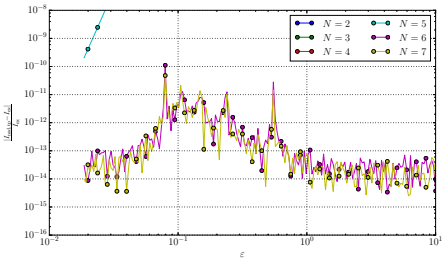
(b) Smolyak construction of level K with a total of $|\Gamma|$ quadrature nodes.



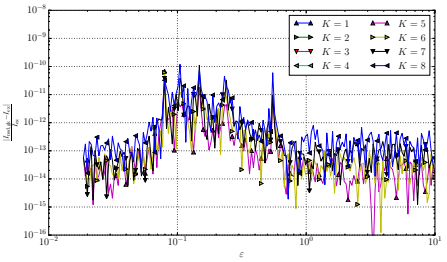
(c) Absolute error of the tensor product ansatz compared to the exact solution.



(d) Absolute error of the Smolyak construction compared to the exact solution.

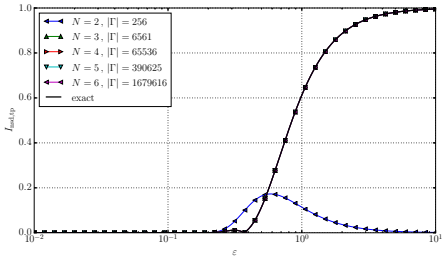


(e) Relative error of the tensor product ansatz compared to the exact solution.

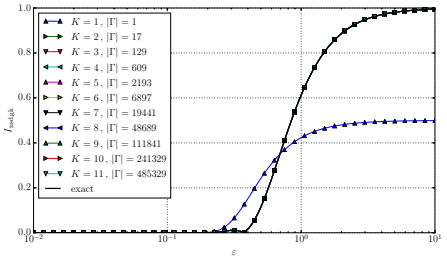


(f) Relative error of the Smolyak construction compared to the exact solution.

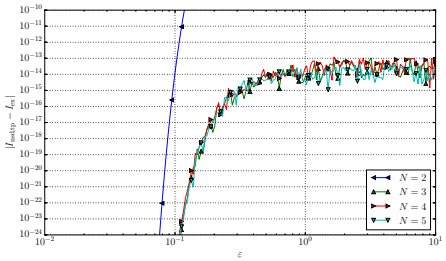
Figure 50: Experiment with $\phi_{5,0,0,0,0,0}$ and $\phi'_{5,0,0,0,0,0}$ in 6 dimensions. The parameters are: $\underline{q} = (1, \ldots, 1)$, $\underline{p} = (-0.2, \ldots, -0.2)$, $\mathbf{Q} = \mathbf{1}$, $\mathbf{P} = \imath\mathbf{1}$ and $\underline{q}' = (1, \ldots, 1)$, $\underline{p}' = (0.2, \ldots, 0.2)$, $\mathbf{Q}' = \mathbf{1}$, $\mathbf{P}' = \imath\mathbf{1}$.



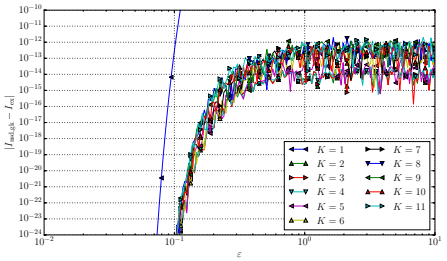
(a) Tensor product of linear size N with a total of $|\Gamma|$ quadrature nodes.



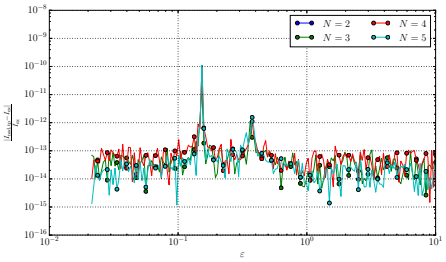
(b) Smolyak construction of level K with a total of $|\Gamma|$ quadrature nodes.



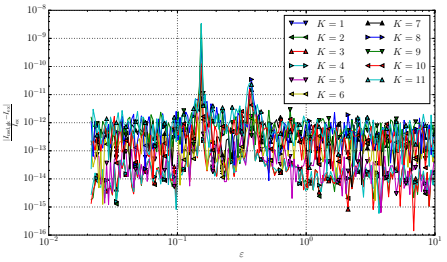
(c) Absolute error of the tensor product ansatz compared to the exact solution.



(d) Absolute error of the Smolyak construction compared to the exact solution.



(e) Relative error of the tensor product ansatz compared to the exact solution.



(f) Relative error of the Smolyak construction compared to the exact solution.

Figure 51: Experiment with $\phi_{2,0,0,0,0,0,0,0}$ and $\phi'_{2,0,0,0,0,0,0,0}$ in 8 dimensions. The parameters are: $\underline{q} = (1, \ldots, 1)$, $\underline{p} = (-0.2, \ldots, -0.2)$, $\mathbf{Q} = \mathbf{1}$, $\mathbf{P} = \imath\mathbf{1}$ and $\underline{q}' = (1, \ldots, 1)$, $\underline{p}' = (0.2, \ldots, 0.2)$, $\mathbf{Q}' = \mathbf{1}$, $\mathbf{P}' = \imath\mathbf{1}$.

9 Conclusion

For the problem of computing overlap integrals of semiclassical wavepackets $|\phi\rangle$ we showed that the common quadrature schemes are insufficient. This is caused by the oscillatory nature of the integrands. Depending on the phase space constellation of the wavepackets involved, Gaussian quadrature can introduce large errors hence entirely destroying the numerical simulation results in the worst case.
Based on recent developments in the area of highly oscillatory quadrature we presented an improvement. This new method adapts the technique of numerical steepest descent specially to the needs of overlap integrals $\langle\phi|\phi'\rangle$. To some extend it is possible also to treat more general operator integrals $\langle\phi|\hat{o}|\phi'\rangle$ but issues may arise from the complex behavior of $\hat{o}$.
The fundamental technique is not restricted to one-dimensional examples and works well in any number of dimensions. There is still a quadrature involved, having a much smaller number of nodes. Full tensor product quadrature becomes expensive in higher dimensions. Therefore we applied the Smolyak construction and used a specially crafted set of nested quadrature points to lessen the curse of dimensionality.
In the future, more work can be done in different directions. Computation for large wavepacket indices k poses an important problem limiting the application of the steepest descent techniques. Being able to compute integrals like $\langle\phi|\hat{o}|\phi'\rangle$ becomes important for some new time-propagation methods currently in development. Finally, we used the classical Smolyak construction while even more sparse schemes like hyperbolic cuts could be beneficent. One might consider adaptive versions too.

References

[1] Andreas Asheim and Daan Huybrechs. Asymptotic Analysis of Numerical Steepest Descent with Path Approximations. *Foundations of Computational Mathematics*, 10:647–671, 2010. 7

[2] Andreas Asheim and Daan Huybrechs. Complex gaussian quadrature for oscillatory integral transforms. *IMA Journal of Numerical Analysis*, 33(4):1322–1341, 2013. 2

[3] Raoul Bourquin. Algorithms for non-adiabatic transitions with one-dimensional wavepackets. 2010. `http://www.sam.math.ethz.ch/~raoulb/research/bachelor_thesis/tex/main.pdf`. 2

[4] Raoul Bourquin. Wavepacket propagation in d-dimensional non-adiabatic crossings. 2012. `http://www.sam.math.ethz.ch/~raoulb/research/master_thesis/tex/main.pdf`. 2

[5] Raoul Bourquin. Exhaustive search for higher-order Kronrod-Patterson extensions. 2014. Unpublished. 46

[6] Raoul Bourquin and Vasile Gradinaru. WaveBlocks: Reusable building blocks for simulations with semiclassical wavepackets. `https://github.com/raoulbq/WaveBlocksND`, 2010 – 2015. 28

[7] Alfredo Deaño and Daan Huybrechs. Complex Gaussian quadrature of oscillatory integrals. *Numerische Mathematik*, 112:197–219, 2009. 2, 18

[8] Alan Genz and Bradley D. Keister. Fully symmetric interpolatory rules for multiple integrals over infinite regions with gaussian weight. *Journal of Computational and Applied Mathematics*, 71(2):299 – 309, 1996. 46

[9] Thomas Gerstner and Michael Griebel. Numerical integration using sparse grids. *Numerical Algorithms*, 18(3-4):209–232, 1998. 46

[10] Andreas Glaser, Xiangtao Liu, and Vladimir Rokhlin. A fast algorithm for the calculation of the roots of special functions. *SIAM Journal on Scientific Computing*, 29(4):1420–1438, 2007. 27

[11] Gene H. Golub and Charles F. Van Loan. *Matrix Computations*. Johns Hopkins University Press, 3rd ed. edition, 1996. 24

[12] George A. Hagedorn. Raising and lowering operators for semiclassical wave packets. *Annals of Physics*, 269(1):77–104, 1998. 2

[13] George A. Hagedorn and Sam L. Robinson. Bohr-Sommerfeld quantization rules in the semiclassical limit. *J. Phys. A*, 31(50):10113–10130, 1998. 2

[14] Florian Heiss and Viktor Winschel. Likelihood approximation by numerical integration on sparse grids. *Journal of Econometrics*, 144(1):62 – 80, 2008. 46

[15] Daan Huybrechs and Stefan Vandewalle. On the Evaluation of Highly Oscillatory Integrals by Analytic Continuation. *SIAM Journal on Numerical Analysis*, 44(3):1026–1048, 2006. 2, 7, 8, 9, 26

[16] Daan Huybrechs and Stefan Vandewalle. The Construction of Cubature Rules for Multivariate Highly Oscillatory Integrals. *Mathematics of Computation*, 76(260):pp. 1955–1980, 2007. 2, 7, 8

[17] Hassan Majidian. Numerical approximation of highly oscillatory integrals on semi-finite intervals by steepest descent method. *Numerical Algorithms*, pages 1–12, 2012. 2, 9, 26

[18] Erich Novak and Klaus Ritter. Simple cubature formulas with high polynomial exactness. *Constructive Approximation*, 15(4):499–522, 1999. 47

[19] B Stefanov, O Iordanov, and L Zarkova. Interaction potential in ${}^1\Sigma_g{}^+ Hg_2$: fit to the experimental data. *Journal of Physics B: Atomic and Molecular Physics*, 15(2):239, 1982. 28

[20] Alex Townsend, Thomas Trogdon, and Sheehan Olver. Fast computation of gauss quadrature nodes and weights on the whole real line. *IMA Journal of Numerical Analysis*, 2015. 27

[21] G.W. Wasilkowski and H. Wozniakowski. Explicit cost bounds of algorithms for multivariate tensor product problems. *Journal of Complexity*, 11(1):1 – 56, 1995. 46

Recent Research Reports

Nr.	Authors/Title

2015-02 R. Hiptmair and S. Sargheini
Scatterers on the substrate: Far field formulas

2015-03 P. Chen and A. Quarteroni and G. Rozza
Reduced order methods for uncertainty quantification problems

2015-04 S. Larsson and Ch. Schwab
Compressive Space-Time Galerkin Discretizations of Parabolic Partial

Differential Equations

2015-05 S. May
New spacetime discontinuous Galerkin methods for solving convection-diffusion systems

2015-06 H. Heumann and R. Hiptmair and C. Pagliantini
Stabilized Galerkin for Transient Advection of Differential Forms

2015-07 J. Dick and F.Y. Kuo and Q.T. Le Gia and Ch. Schwab
Fast QMC matrix-vector multiplication

2015-08 P. Chen and Ch. Schwab
Adaptive Sparse Grid Model Order Reduction for Fast Bayesian Estimation and Inversion

2015-09 J.-L. Bouchot and B. Bykowski and H. Rauhut and Ch. Schwab
Compressed Sensing Petrov-Galerkin Approximations for Parametric PDEs

2015-10 A. Jentzen and P. Pusnik
Strong convergence rates for an explicit numerical