JOURNAL OF COMPUTATIONAL PHYSICS 100, 364–376 (1992)

Comparison of Coordinate-Space Techniques in Nuclear Mean-Field Calculations

V. BLUM

Institut für Theoretische Physik der Universität Frankfurt a.M., Robert Mayer-Strasse 10, 6000 Frankfurt a.M. 11, Germany

G. LAURITSCH

Institut für Theoretische Physik der Universität Erlangen/Nürnberg, Staudtstrasse 7, 8520 Erlangen, Germany

J. A. MARUHN

Institut für Theoretische Physik der Universität Frankfurt a.M., Robert Mayer-Strasse 10, 6000 Frankfurt a.M. 11, Germany

AND

P.-G. REINHARD

Institut für Theoretische Physik der Universität Erlangen/Nürnberg, Staudtstrasse 7, 8520 Erlangen, Germany

Received November 1, 1990; revised June 20, 1991

We investigate three different numerical representations for nuclear mean-field calculations: finite differences, Fourier representation, and basis-splines. We compare these schemes with respect to precision and speed. It turns out that Fourier techniques and basis-splines are much superior in precision to finite differences. The Fourier representation in connection with the fast Fourier transformation is advantageous for large grids whereas matrix techniques, derived either from basis-splines or from Fourier representation, are preferable for smaller grids. © 1992 Academic Press, Inc.

1. INTRODUCTION

Mean-field calculations have been widely used in nuclear physics for two decades. They are the many Hartree–Fock, deformed Hartree–Fock, and time-dependent Hartree–Fock (TDHF) calculations based on the nonrelativistic Skyrme force model [1, 2]. And there seem to be still several open questions which deserve further investigation in these models [3]. There are, on the other hand, similar calculations based on the relativistic mean-field model [4–6]. A representation of the wavefunctions on a coordinate-space (or momentum-space) grid is used in a dominant fraction of the codes for these mean-field calculations. This representation is preferable because it provides a very easy and transparent way of programming the Schrödinger equation. It is also a very efficient technique and it is well adapted for vectorization.

The fastest and simplest coordinate-space technique is the representation of the second derivatives in the kinetic energy operator by a three-point finite difference formula. An optimum within the three-point precision can be achieved from variation on the grid, i.e., variation of the discretized action [7]. This technique has been applied successfully to nonrelativistic nuclear TDHF models with [3] and without [8] spin-orbit force. And it has been also applied in relativistic deformed Hartree–Fock calculations [9]. However, the relativistic models have revealed most clearly the severe limitations of the simple finite difference approaches with three-point precision [10]. The reason for the deficiencies is the spatial variation of the effective mass in the kinetic energy. These variations are particularly demanding concerning numerical precision. The problems, however, are also present in nonrelativistic models of nuclear structure. A further problem occurs in relativistic calculations where the three-point formula induces "fermion-doubling" [12]. Thus one needs in any case to improve the numerical representation. But this should be done in a way which preserves the simplicity of a coordinate- (or momentum-) space representation and which keeps the numerical expenses low.

We will study three alternatives to the three-point finite difference formula: first, finite difference schemes with higher order, second, a momentum-space representation with the fast Fourier transformation (FFT) to connect coordinate- with momentum-space [11]; and, third, a

0021-9991/92 $5.00

representation of the wavefunction in terms of basis-splines (B-splines), which looks like a coordinate-space representation as local operators are concerned and which uses matrix multiplication to represent derivatives [12]. The FFT techniques have been used for a long time in large scale nuclear TDHF calculations as well as in the relativistic ones [13]. The B-splines have been presented extensively in Ref. [14]. An application for solving the Dirac equation is presented in [12]. And meanwhile they are also being used for handling nonrelativistic TDHF with spin-orbit force [15].

It is the aim of this paper to provide a critical comparison of these three methods with respect to numerical precision and efficiency. We consider a Hamiltonian which arises typically in nuclear mean-field calculations. It is distinguished by a strongly varying effective mass in kinetic energy. We also briefly consider models with constant mass, as occur in simple nuclear models or in effective density functionals for atomic physics calculations [16]. Finally we look at a special form of relativistic models, where spin and momentum are contracted to a scalar which can possibly simplify the handling. It will turn out that the choice of the optimum numerical representation depends very much on the particular application and to a certain extent on the computer system used. Therefore the typical Hamiltonians as well as the typical methods for static and dynamic calculations are outlined and all detailed material for a decision on the optimal method in a given case are provided. This has the welcome side effect that the present paper is also a short summary of numerical techniques in Hartree–Fock and time-dependent Hartree–Fock calculations. The considerations on efficiency and speed have been based on actual test runs on two vector machines, an IBM 3090 VF and a CRAY X-MP. We thus hope also to give some idea of the machine dependence of a decision for a numerical scheme.

The paper is outlined as follows: in Section 2 we present the typical Hamiltonians and give a survey of the possible solution schemes for static and dynamic calculations. In Section 3, we list the various dimensionalities which arise in practice. In Section 4, we give a short survey of the various gridding techniques in coordinate- and momentum-space. In Section 5, we discuss the question of precision of the various gridding techniques. And in Section 6, we estimate and compare the speed of the various methods extensively.

2. THE TYPICAL APPLICATIONS

2.1. The Hamiltonian

The most general mean-field Hamiltonian in nuclear physics applications has the form

$$h = \mathbf{p}B(\mathbf{r})\cdot\mathbf{p} + \mathbf{W}(\mathbf{r})\cdot(\mathbf{p}\times\boldsymbol{\sigma}) + V(\mathbf{r}), \tag{1}$$

where the potential V, the spin-orbit potential $\mathbf{W}$, and the inverse effective mass B all depend on the occupied single-particle states. It is our experience that this self-consistency has no effect on the numerical stability within the iterative schemes discussed below. Therefore we can consider for the purpose of our tests an external field problem, where the potentials and masses are readily given as in a nuclear shell model. We shape the fields to have a Saxon–Woods form with the proper shell oscillations on top [17]:

$$\begin{aligned} V &= V_0 \frac{1 + A_V \cos(2k_F r)}{1 + \exp((r-R)/\sigma)}, \\ B &= \frac{\hbar^2}{2m} + B_0 \frac{1 + A_B \cos(2k_F r)}{1 + \exp((r-R)/\sigma)}, \\ \mathbf{W} &= 0 \end{aligned} \tag{2}$$

The parameters of the model potential (2) have been chosen to simulate a Hartree–Fock Hamiltonian in a small nucleus:

$$\begin{aligned} &V_0 = -70.0\ \text{MeV}, && A_V = 0 \\ &B_0 = \hbar^2/2m, && A_B = -0.5 \\ &R = 2.0\ \text{fm}, && \sigma = 0.3\ \text{fm}, \qquad k_F = 1.35\ \text{fm}^{-1}. \end{aligned} \tag{3}$$

The effective mass $\hbar^2/2m/B$ is particularly small (≈ 0.5), the shell oscillations (parameter A_B) are very large, and the surface thickness σ is extremely small in order to provide a most critical test of the numerical representations. The spin-orbit force W has been neglected in the present consideration. It has little effect on the precision compared to the most critical strong variations of the effective mass.

In some cases, one considers less demanding mean-field Hamiltonians. The simplest approach is to neglect the effective mass and to use the constant nucleon mass. This yields the approach

$$h = \frac{\hbar^2}{2m}\mathbf{p}^2 + V(\mathbf{r}). \tag{4}$$

An example for this kind of schematic force is the Bonche–Koonin–Negele force which has been invented and used many times as a schematic model for nuclear TDHF dynamics [18]. The form (4) is also appropriate in atomic mean-field models as, e.g., for the calculation of liquid metal clusters [19] using the Lundquist energy functional [16].

Spin and momentum are particularly connected in simple relativistic models such that the effective mean-field Hamiltonian becomes [6]

$$h = \boldsymbol{\sigma}\cdot\mathbf{p}B(\mathbf{r})\,\boldsymbol{\sigma}\cdot\mathbf{p} + V(\mathbf{r}), \tag{5}$$

where B and V behave typically as given in Eq. (2). The

scalar contraction of $\boldsymbol{\sigma}$ and $\mathbf{p}$ can simplify the evaluation in the momentum-representation, and moreover, the spin-orbit force $\mathbf{W} = \nabla B$ is already included in the Hamiltonian (5) at no extra expense.

2.2. Damped Gradient Iteration for the Static Case

In the static case, one has to solve the stationary Hartree–Fock equation

$$h\psi_\alpha = \varepsilon_\alpha \psi_\alpha, \tag{6}$$

where h depends implicitly on all occupied single particle states ψ_α. It is most efficiently solved with the damped gradient iteration [20]

$$\psi_\alpha^{(n+1)} = \mathcal{O}\{[1 - \mathcal{D}(h - \langle h \rangle)]\, \psi_\alpha^{(n)}\} \tag{7}$$

$$\mathcal{D} = \frac{x_0}{(\hbar^2/2m)\,\mathbf{p}^2 + E_0}, \tag{8}$$

where $\langle h \rangle$ is the expectation value of h and $\mathcal{D}$ is the kinetic energy damping with numerical parameters x_0 and E_0 to be chosen such that stable and fast convergence is achieved. The choice depends on the actual system and one needs to do a few experiments in order to find the optimum. Reasonable starting values are $x_0 \approx 0.8$ and $E_0 \approx -0.5 V_{\min}$ with $V_{\min}$ being the minimum of $V(\mathbf{r})$. The $\mathcal{O}$ means orthonormalisation of all occurring single particle wave functions. In practice, we use Gram–Schmidt orthonormalisation. Thus the static step consists of three substeps: first, Hamiltonian action $h\psi$; second, damping $\mathcal{D} \propto 1/(\mathbf{p}^2 + \text{const})$; and third, orthonormalization.

The damped gradient iteration (7), (8) can also be applied to constrained Hartree–Fock calculations which are used to construct deformation-energy surfaces [21]. Thus the considerations of this paper apply to a very wide range of microscopic models.

2.3. Time Steps for TDHF

In the dynamic case, one has to solve the TDHF equation

$$i\,\partial_t \psi_\alpha(t) = h\psi_\alpha(t). \tag{9}$$

The problem here is that there is a wider range of strategies for the time step. Predictor-corrector schemes have often been used in connection with the FFT representation [11], the simplest explicit Euler step has also been used under the name imaginary-time method, or straightforward power-law expansions of the time-evolution operator $e^{-ih(t-t_0)}$ are considered as improvements of the imaginary-time method [22]. Here we consider the Crank–Nicholson method as the most promising and most typical time step for the TDHF propagation,

$$\psi_\alpha(t + \delta t) = \left(1 + \frac{i}{2}\,\delta t h\right)^{-1} \left(1 - \frac{i}{2}\,\delta t h\right) \psi_\alpha(t), \tag{10}$$

where h is to be taken at half the time step $h = h(t + \delta t/2)$. This requires an extra trial shot to $\psi_\alpha(t + \delta t/2)$. One may alternatively consider $h = \frac{1}{2}(h(t) + h(t + \delta t))$ of one is going to compute the inverse in step (10) iteratively anyway. Note that there is no orthonormalization required in the Crank–Nicholson step because the propagation operator in Eq. (10) is manifestly unitary and this automatically guarantees proper orthonormalization of the propagated wave functions.

The inversion is comparable to the damping in the static case, except for the fact that the full mean-field Hamiltonian h is used here. The inversion is still feasible in one dimension. In higher dimensions, one may either use an iteration scheme,

$$\psi = A^{-1}\varphi : \psi^{(n+1)} = \psi^{(n)} - \mathcal{D}[A\psi^{(n)} - \varphi], \tag{11}$$

with the damping $\mathcal{D}$ as given in Eq. (8) and with $A = 1 + (i/2)\,\delta t h$. Or, as a cheaper alternative, one may use separable approximations to the inversion. The possible separability depends on the actual situation. It could read, for example, in three dimensions and without spin-orbit force

$$\left(1 + \frac{i}{2}\,\delta t h\right)^{-1} = \prod_{i=x,y,z} \left(1 + \frac{i}{2}\,\delta t h_i\right)^{-1}, \qquad h_i = \hat{p}_i B \hat{p}_i + \tfrac{1}{3} V \tag{12}$$

which factorizes the full inversion into three successive one-dimensional steps.

3. THE VARIOUS DIMENSIONS

The various dimensionalities considered are:

1D cartesian. For this case we have $\mathbf{r} \to x$, $\psi(\mathbf{r}) \to \psi(x)$, and the Hamiltonian becomes simply

$$h = \hat{p}_x B(x)\, \hat{p}_x + V(x). \tag{13}$$

The inversion in the damped gradient step (7) and in the Crank–Nicholson step (10) can be done exactly by inversion of the corresponding matrix.

1D spherical. We now have three-dimensional dynamics, but reduced by symmetry to $\psi(\mathbf{r}) \to (R_l(r)/r)\, Y_{lm}(\theta, \phi)$. The radial Hamiltonian and the

damping operator for the reduced wave function $R_l(r)$ is then simply

$$h = \hat{p}_r B(r)\, \hat{p}_r + B(r)\frac{l(l+1)}{r^2} + V(r), \qquad \hat{p}_r = -i\frac{d}{dr} \tag{14}$$

$$\mathscr{D} = x_0 \Big/ \left(\frac{\hbar^2}{2m}\hat{p}_r^2 + \frac{\hbar^2}{2m}\frac{l(l+1)}{r^2} + E_0\right). \tag{15}$$

Note that the Hamiltonian and the damping operator depend on the angular momentum of the wavefunction. The inversions can be done as in the previous case.

2D axial. In that case, the three-dimensional dynamics is reduced only to $\psi(\mathbf{r}) \to \phi_m(r, z)\exp(-im\phi)$. The Hamiltonian and the damping for the $\phi_m(r, z)$ then are

$$h = \hat{p}_z B(r, z)\, \hat{p}_z + \frac{1}{r}\hat{p}_r(rB(r, z)\, \hat{p}_r) + B(r, z)\frac{m^2}{r^2} + V(r, z) \tag{16}$$

$$\mathscr{D} = x_0 \Big/ \left(\frac{\hbar^2}{2m}\left(\hat{p}_r^2 + \frac{1}{r}\hat{p}_r + \frac{m^2}{r^2}\right) + \frac{\hbar^2}{2m}\hat{p}_z^2 + E_0\right). \tag{17}$$

The inversion is now complicated by the fact that the full matrix in $r-z$-space is rather large. One can simplify the inversion of the damping operator by a separable approach:

$$\begin{aligned} \mathscr{D} &\approx \frac{x_0}{E_0}\mathscr{D}_r\mathscr{D}_z, \\ \mathscr{D}_r &= 1\Big/\left(\frac{\hbar^2}{2mE_0}\left(\hat{p}_r^2 + \frac{1}{r}\hat{p}_r + \frac{m^2}{r^2}\right) + 1\right), \\ \mathscr{D}_z &= 1\Big/\left(\frac{\hbar^2}{2mE_0}\hat{p}_z^2 + 1\right). \end{aligned} \tag{18}$$

The inversion in the Crank–Nicholson step (10) may be separated similarly if there is no spin-orbit force [8]. One has to go up to a fivefold separation in cases with spin-orbit force [3]. There may be situations where it is more efficient to use the exact inversion in the Crank–Nicholson step. Then one may use the iterative inversion (11) with the separable approach (18) for the damping therein.

3D cartesian. The kinetic energy separates in the three directions. The Hamiltonian and the damping then are

$$h = \hat{p}_x B(\mathbf{r})\, \hat{p}_x + \hat{p}_y B(\mathbf{r})\, \hat{p}_y + \hat{p}_z B(\mathbf{r})\, \hat{p}_z + V(\mathbf{r}) \tag{19}$$

$$\mathscr{D} = x_0 \Big/ \left(\frac{\hbar^2}{2m}(\hat{p}_x^2 + \hat{p}_y^2 + \hat{p}_z^2) + E_0\right). \tag{20}$$

The inversion can be done approximately with a separation similarly to the 2D axial case:

$$\begin{aligned} \mathscr{D} &\approx \frac{x_0}{E_0}\mathscr{D}_x\mathscr{D}_y\mathscr{D}_z, \\ \mathscr{D}_i &= 1\Big/\left(\frac{\hbar^2}{2mE_0}\hat{p}_i^2 + 1\right), \qquad i \in \{x, y, z\}. \end{aligned} \tag{21}$$

A sevenfold separation comes into play if the spin-orbit force is included.

4. THE GRIDDING TECHNIQUES

There are essentially three different ways of representing a wavefunction numerically: either on a grid in coordinate-space, or on a grid in momentum-space, or in a basis of analytically given functions (e.g., a basis of harmonic oscillator functions is often used). The coordinate- or momentum-space representations with equidistant grids have turned out to be very efficient in saturating systems, as, e.g., nuclei, metallic clusters, or liquid ^{3}He. They have the additional advantage of providing a straightforward programming style and simple vectorization. We thus concentrate the following considerations to coordinate- and momentum-space representations. In the following we will briefly present the typical gridding techniques.

4.1. Coordinate-Space Representation

The wave function is stored and handled on a discretized and finite grid in coordinate-space, i.e.,

$$\psi = \psi(\mathbf{r}_i), \tag{22}$$

where i labels the grid-points. We will only discuss equidistant grids. This reads, e.g., for one spherical dimension, $r_i = i\,\Delta r$, $i = 0, ..., N$, and similarly for the other dimensions.

The action of the local operators $V(\mathbf{r})$ and $B(\mathbf{r})$ is a simple multiplication in coordinate-space representation, e.g., $(V\psi)(\mathbf{r}_i) = V(\mathbf{r}_i)\,\psi(\mathbf{r}_i)$. All the complicated work comes in with the momentum $\hat{p}_v$, where v labels here, and in the following, one of the components x, y, z, or r, depending on the dimension of the problem. We can represent it in general as a matrix in the coordinate-space grid

$$(\hat{p}_v\psi)(\mathbf{r}_i) = \sum_j (\mathscr{P}_v)_{ij}\,\psi(\mathbf{r}_j). \tag{23}$$

The action of the Hamiltonian is then

$$h\psi = \sum_v \mathscr{P}_v B \mathscr{P}_v \psi + V\psi. \tag{24}$$

It obviously requires two matrix multiplications per spatial dimension if the inverse effective mass B depends on $\mathbf{r}$. The number of matrix multiplications remains the same if we use the alternative form for the Hamiltonian,

$$h\psi = B\sum_v \mathscr{P}_v^2\psi + \sum_v (\mathscr{P}_v B)(\mathscr{P}_v\psi) + V\psi. \qquad (25)$$

The form (25) is advantagous for poor representations of $\mathscr{P}$, as, e.g., finite differences of third order, and if B does not vary too much. The more symmetric form (24) is generally to be preferred.

There are many variants of coordinate-space representations which are distinguished by different schemes to approximate the momentum operator $\mathscr{P}$. Some of them will be presented in the following subsections.

4.1.1. *Finite Difference Schemes*

The standard approach to $(\mathscr{P}_v)_{ij}$ is to use the well-known finite difference formulas of order 3, 5, 7, etc. for d/dr_v or d^2/dr_v^2; see, e.g., [23]. It is to be noted that the finite difference formula for d/dr_v is less precise than the finite difference formula for d^2/dr_v^2 of the same order. Thus one should use the form (25) for the Hamiltonian and carry two sparse matrices, one for $(\mathscr{P}_v^2)_{ij}$ and one for $(\mathscr{P}_v)_{ij}$. This is clearly necessary for three-point precision, it is quantitatively advantagous for five-point precision, and it is a matter of taste for seven-point and higher precision.

The advantage of the finite difference schemes is that one obtains sparse matrices for $\mathscr{P}_v$ or $\mathscr{P}_v^2$. The expense then grows only linearly with one spatial dimension. However, one should not increase the order arbitrarily because the usual finite difference formulas become increasingly sensitive to noise in the numerical data. Using fifth order is still a fairly robust scheme.

4.1.2. *B-splines*

A more robust representation of the derivative matrix is obtained from using B-splines [12, 14]. One starts with a representation of the wavefunctions and potential fields as superpositions of B-splines. The B-splines can be applied to differential equations either using the Galerkin method, which in this case corresponds to a minimization of the expectation value of the Hamiltonian, or in the collocation approximation, in which the differential equations themselves are solved on a set of *collocation points* in space. The Galerkin method, corresponding essentially to a basis expansion in the full set of B-Splines in all three coordinate directions, cannot easily be decomposed into one-dimensional matrix manipulations, while, as shown below, this is possible for the collocation method, so that the latter will be studied exclusively in this paper. In this case a local representation in coordinate-space emerges, where now the derivative matrix $\mathscr{P}_{ij}$ is a fully packed matrix for each order of the B-splines used. (This occurs because a matrix inversion is involved.) Furthermore, the robustness of the B-splines allows increasing the order arbitrarily, without running into problems with numerical noise. Thus it is advisable to use representations of high orders. The high precision of the derivatives $\mathscr{P}_{ij}$ then allows using the more symmetric form (24) for the Hamiltonian. The local operators V and B are handled as simple local products in coordinate-space. The collocation approximation limits the precision of the representation as we will see in the explicit tests later.

4.1.3. *Fourier Definition of the Momentum*

A well-established, very robust, and the most precise definition of the momentum on a grid is the Fourier representation,

$$\hat{p}_v = \mathscr{F}_v^{-1}\hbar k_v \mathscr{F}_v, \qquad v \in \{x, y, z\}. \qquad (26)$$

The $\mathscr{F}_v$ is the Fourier transformation in the v-direction and k_v stands for a simple multiplication in Fourier space. $\mathscr{F}_v^{-1}$ will denote the backward transform in the following. The Fourier definition of $\hat{p}_v$ can be handled in two ways. Either one uses the form (26) directly which means to perform forward and backward Fourier transformations explicitly, whenever $\hat{p}_v$ is required, or to combine the operations into the matrix $(\mathscr{P}_v)_{ij}$ once and forever; i.e., one initializes $\mathscr{P}_v$ by $(\mathscr{P}_v)_{ij} \equiv \mathscr{F}_v^{-1}\hbar k_v \mathscr{F}_v$ for each v-direction required. In the latter case, one obtains a Fourier definition of the momentum matrix $(\mathscr{P}_v)_{ij}$, but the further handling of the momentum as a matrix is technically identical to the approach using $(\mathscr{P}_v)_{ij}$ generated from B-splines. The difference is only in the setup of the matrix $(\mathscr{P}_v)_{ij}$.

4.2. Momentum-Space Representation

One can also handle and store the wave functions in momentum-space, i.e., $\psi \equiv \tilde{\psi}(k)$. Coordinate- and momentum-space are connected by complex Fourier transformations. One may reduce these to the simpler sin- or cos-transformations if the wavefunctions have well-defined reflection symmetry about $x = 0$. In any case, the Fourier transformation is numerically very efficient to handle with the well-known fast Fourier transformation (FFT). The action of $\hat{p}$ becomes a simple product with $\hat{p} \equiv k$ in momentum-space. The local operators are the costly part in the momentum-space representation. They are evaluated as $\mathscr{F} V(r) \mathscr{F}^{-1}$ and similarly for B. Thus they require one forward and one backward Fourier transform.

The analogous transformation for the axially symmetric case is the Bessel transformation, $\tilde{\psi}_m(k) = \int dr r J_m(kr) \times \psi_m(r)$, where m is the angular momentum along the z-axis. And for the spherical radial coordinate it reads

$\tilde{\psi}_l(k) = \int dr r^2 j_l(kr)\,\psi_l(r)$, where l is the orbital angular momentum and the j_l are the spherical Bessel functions. There are no fast schemes for these two transformations. Each transformation counts as one full matrix multiplication. Therefore, the momentum-space representations are not very useful in the axial or spherical case.

The Fourier basis gives an arbitrarily precise representation of the kinetic energy. The maximum kinetic energy for a given grid size is much higher than in other comparable griddings and the high end of the spectrum is rarely exploited. Thus one may cut off the wave functions in momentum-space, see Fig. 2 in the next section. As a rule of thumb, a cutoff in $|\mathbf{k}|$ at *half the maximum momentum* is usually a good compromise between speed and precision and gives a dramatic reduction to 7% of the phase space in the 3D cartesian case. Note that this reduction requires a special storage algorithm (pointers) for the wave function in the three-dimensional momentum-space.

5. THE TEST OF THE PRECISION

In a first step, we have investigated the precision of the various representations as function of the stepsize. The testcase was the Hamiltonian (1) with the potentials (2) for radial symmetry and dependence on the radius only (1D spherical). The potential parameters are given in Eq. (3). In Fig. 1 we show the single particle energy of the $1s_{1/2}$ state in the chosen spherical model as function of the grid spacing Δr for the various representations discussed above. We see that the usual third-order finite difference scheme yields a terribly bad precision. One can easily gain substantial improvements by increasing the order of the finite difference scheme. Fifth order already provides a fair approach if one uses small enough grid spacing Δr. Higher order would be desirable. But then one may run into problems with stability in large scale applications because the higher order finite difference formulas become increasingly sensitive to numerical noise in the wavefunctions. Thus the preferable choice with finite differences is to use a fifth-order scheme and small enough grid spacing Δr.

The third-order B-spline representation is as poor as the third-order finite difference scheme. But the higher orders improve faster, with final precision reached already at seventh order. Note that it is unproblematic with the B-splines to go to every desired order because the scheme is robust against noise and one needs to handle full matrices anyway. One may be a bit surprised that the precision ceases to improve after the seventh order. The reason for this behaviour is the collocation approximation which limits the precision independent of the quality of the kinetic energy. Every further improvement with B-splines would require nonlocal representations for potential fields. One should keep in mind this option for further numerical development.

The Fourier representation gives very good precision already at fairly large grid spacings, Δr. The quality is of the order of the best B-splines or even slightly above. A feature which has not been visualized in Fig. 1 is that the precision of the Fourier representation is independent of a cutoff in momentum-space (Fig. 2) down to kinetic energy $\hbar^2k^2/2m = 150$ MeV which corresponds to half the maximum momentum for $\Delta r = 0.6$ fm. Thus the kinetic energy could live very well with a grid spacing $\Delta r \approx 1.2$ fm, but again the local potentials limit the precision, similar to the case of the B-splines.

The comparison of precision based on Fig. 1 has shown poor precision for the finite difference schemes, and equally high precision for both the Fourier representation and the B-splines. It is necessary to discuss the actual computing times in order to estimate the efficiency of the schemes.

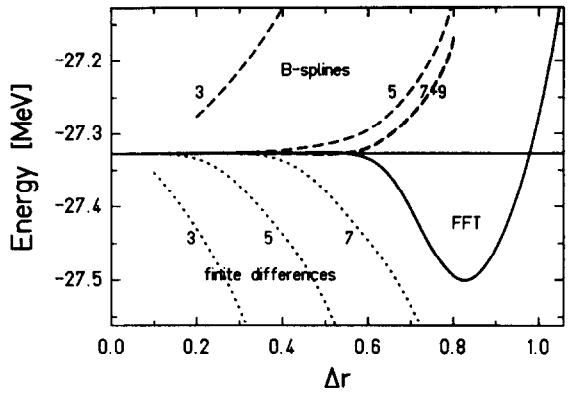

FIG. 1. Single particle energy of the $1s_{1/2}$ state from the Hamiltonian (1), (2) with parameters (3) as function of the radial grid spacing Δr for various gridding techniques: finite differences (dotted lines) of various orders as indicated; B-splines (dashed lines) of various orders as indicated; and Fourier representation (full line) indicated by the label FFT.

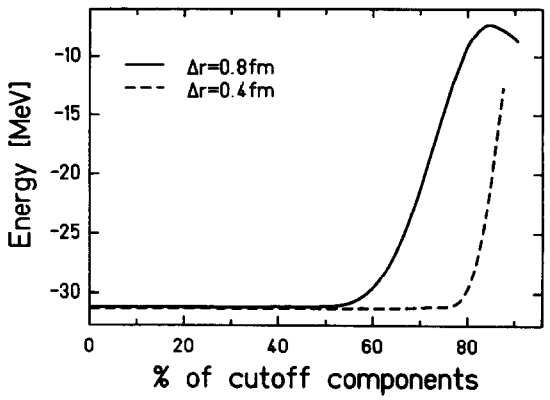

FIG. 2. The energy of the $1s_{1/2}$ state from the Hamiltonian (1), (2) with parameters (3) in dependence on the cutoff in momentum-space for two grid-sizes as indicated.

6. ESTIMATES OF COMPUTING TIMES

The best test of computing times is, of course, to write a code for each scheme in full dimensionality. This is very cumbersome. The other extreme is a simple estimate by counting floating point operations. This is too simplistic. We have decided to follow an intermediate procedure: We first evaluate realistic computing times for the basic operations, i.e., matrix multiplication or FFT in one dimension, and then evaluate the required number of these basic operations in higher dimensions. This allows us to take into account, to some extent, vectorization features and the gain by using optimized system routines of a computer. This is, of course, still a rough estimate because extrapolating results from one dimension to more dimensional cases corresponds to straightforward programming. In higher dimensions, it may very well be possible to obtain additional gains in speed by rearranging loops and adjusting the computed partitions to cache-size or register-size, etc. It is obvious that such a comprehensive comparison of the best optimized codes for each scheme and, in each case, of dimensions and Hamiltonians is a cumbersome and, perhaps, useless task. What we need is an a priori estimate of the optimum method before starting the very lengthy and detailed setup of a large-scale program and its optimization. Thus our working hypothesis is: not knowing what the possible advantages of hand-optimizing for the various schemes are, we assume equidistribution of the gains and live with the simple estimate as given in the following.

One sees from the above discussion that an estimate of computing times becomes machine dependent. We will discuss in the following two mainframe computers which both provide vectorizations, namely an IBM 3090 with vector unit and a CRAY X/MP. Both machines are used with one processor only.

6.1. Computing Times for the Basic Operations

We have compared the computing times for the FFT, for full matrix multiplication, and for the sparse matrix multiplication of the finite difference scheme of fifth order. The result is shown in Table I. Although the table serves mainly as input for the estimates of expenses in the following section, a few interesting features may be seen directly here: The expense of the FFT, $\mathscr{E}_{\mathrm{FFT}}$, indeed grows slower with N than the expense of the matrix operation, $\mathscr{E}_{\mathrm{Matr}}$, as expected. But note that the FFT is not faster than matrix multiplication at low N. The fifth-order finite difference matrix behaves similarly in that it does not yield much of an advantage over the full matrix multiplication at low N. But it becomes substantially faster than any other scheme at very high N.

It is to be noted that the system routine for multiplication with sparse matrices on the CRAY is much slower than a simple compiled Fortran source code. Thus we have entered the more favourable case for the option FD5 in Table I. We always will use the more optimal routines in the composed estimates of expense in the subsequent subroutines. This means that we use system routines in general, except for the finite differences (FD5) on the CRAY, where we use the faster compiled code.

There are further interesting differences between the two machines. The IBM seems to optimize the FFT better than the CRAY, at the level of the system routines. This hints at different vectorization features in the two machines. The relation of system routines to user-written routines seems to be more favourable for the CRAY. We do not know whether this is due to a better optimization of system routines on the CRAY or a better optimizing compiler on the IBM.

As a note aside, we always have used full matrix multiplication or full complex Fourier transformation. One may expect some gain in speed by exploiting hermiticity of the momentum matrices or symmetry- and reality-restrictions in the Fourier transformation. The problem is that system routines for multiplication with symmetric matrices have not been readily available in all cases. And negligible gain (if not loss) was observed in those cases where we could use symmetry. Very probably we again would run into problems with optimization and vectorization which would require a very detailed hand-optimizing of the code. Thus we did not try to include symmetry considerations into our test, again recalling our working hypothesis that the possible gains will be equidistributed amongst all the methods.

TABLE I

Computing Times $\mathscr{E}$ in CPU Seconds for 10,000 Repeated Operations $\mathscr{F}\psi$ for $\mathscr{E}_{\mathrm{FFT}}$ or $\mathscr{P}\psi$ for $\mathscr{E}_{\mathrm{FD5}}$ and $\mathscr{E}_{\mathrm{Matr}}$ with Varying Grid Size N and for a Complex Wavefunction ψ

	IBM 3090 VF					CRAY X-MP				
	System routine			Compiled code		System routine		Compiled code		
N	$\mathscr{E}_{\mathrm{FFT}}$	$\mathscr{E}_{\mathrm{FD5}}$	$\mathscr{E}_{\mathrm{Matr}}$	$\mathscr{E}_{\mathrm{FFT}}$	$\mathscr{E}_{\mathrm{Matr}}$	$\mathscr{E}_{\mathrm{FFT}}$	$\mathscr{E}_{\mathrm{Matr}}$	$\mathscr{E}_{\mathrm{FFT}}$	$\mathscr{E}_{\mathrm{FD5}}$	$\mathscr{E}_{\mathrm{Matr}}$
8	0.53	0.34	0.42	0.5	0.28	0.12	0.06	0.52	0.07	0.66
16	0.63	0.38	0.62	1.0	0.46	0.14	0.11	0.96	0.07	1.32
32	0.92	0.42	1.18	2.0	1.02	0.21	0.28	1.80	0.08	2.97
64	1.17	0.52	2.86	4.5	2.60	0.43	0.92	3.63	0.09	8.69
128	1.86	0.74	12.30	9.0	160.00	1.15	3.73	7.74	0.13	25.93
256		1.18				3.50	14.41	17.18	0.21	82.36

Note. The index FFT explains itself, the index FD5 means the finite difference scheme of fifth order and the index "Matr" means full matrix multiplication which applies for the B-spline representation as well as a Fourier representation of the matrix $\mathscr{P}$. "System routine" means that an optimized subroutine from the system library of each machine was used, whereas "compiled code" means code generated from compiling Fortran source code using the optimizing compiler on each machine.

6.2. 3D Cartesian

6.2.1. *Action of the Hamiltonian*

The Fourier transformation for a three-dimensional wavefunction $\psi(x, y, z)$ into the transformed $\tilde{\psi}(k_x, k_y, k_z)$ is separated in three successive transformations in each one of the three directions. This means that a three-dimensional Fourier transform requires

$$\mathscr{E}_{\mathscr{F}_3}(N_x, N_y, N_z) = N_x N_y \mathscr{E}_{\mathrm{FFT}}(N_z) + N_z N_x \mathscr{E}_{\mathrm{FFT}}(N_y) + N_y N_z \mathscr{E}_{\mathrm{FFT}}(N_x), \tag{27}$$

where $\mathscr{E}_{\mathrm{FFT}}(N)$ is the expense of one-dimensional FFT as given in Table I. The action of the Hamiltonian (19) separates in the three derivatives:

$$h\tilde{\psi} = k_x \mathscr{F}_3^{-1} B \mathscr{F}_3 k_x \tilde{\psi} + k_y \mathscr{F}_3^{-1} B \mathscr{F}_3 k_y \tilde{\psi} + k_z \mathscr{F}_3^{-1} B \mathscr{F}_3 k_z \tilde{\psi} + \mathscr{F}_3^{-1} V \mathscr{F}_3 \tilde{\psi}. \tag{28}$$

We neglect the trivial actions of B or V in coordinate-space and of k_y in momentum-space, so that the expense for one Hamiltonian step with variable inverse mass B is

$$\mathscr{E}_{\mathrm{FFT},B,3\mathrm{D}}(N_x, N_y, N_z) = 8\{N_x N_y \mathscr{E}_{\mathrm{FFT}}(N_z) + N_z N_x \mathscr{E}_{\mathrm{FFT}}(N_y) + N_y N_z \mathscr{E}_{\mathrm{FFT}}(N_x)\}. \tag{29}$$

The situation is much more favourable for a constant inverse mass, for which the kinetic energy becomes $(\hbar^2/2m)\,k^2$ and can be evaluated directly in momentum-space and it remains for the expense

$$\mathscr{E}_{\mathrm{FFT},m,3\mathrm{D}}(N_x, N_y, N_z) = 2\{N_y N_z \mathscr{E}_{\mathrm{FFT}}(N_x) + N_z N_x \mathscr{E}_{\mathrm{FFT}}(N_y) + N_x N_y \mathscr{E}_{\mathrm{FFT}}(N_z)\} \tag{30}$$

which comes completely from the action of $V(r)$.

In coordinate-space identical considerations lead to

$$h\psi = \mathscr{P}_x B \mathscr{P}_x \psi + \mathscr{P}_y B \mathscr{P}_y \psi + \mathscr{P}_z B \mathscr{P}_z \psi + V\psi \tag{31}$$

and

$$\mathscr{E}_{\mathscr{P},B,3\mathrm{D}}(N_x, N_y, N_z) = 2\{N_y N_z \mathscr{E}_{\mathscr{P}}(N_x) + N_z N_x \mathscr{E}_{\mathscr{P}}(N_y) + N_x N_y \mathscr{E}_{\mathscr{P}}(N_z)\} \tag{32}$$
$$\mathscr{E}_{\mathscr{P}} \in \{\mathscr{E}_{\mathrm{FD5}}, \mathscr{E}_{\mathrm{Matr}}\}$$

because the FFTs are effectively replaced by the corresponding matrix multiplications. $\mathscr{E}_{\mathscr{P}}$ includes both choices for $\mathscr{P}$, the full matrix case $\mathscr{E}_{\mathrm{Matr}}$, or the finite difference scheme $\mathscr{E}_{\mathrm{FD5}}$. Again, the case with constant mass allows some simplification and the expense is reduced by a factor of two to

$$\mathscr{E}_{\mathscr{P},m,3\mathrm{D}}(N_x, N_y, N_z) = N_y N_z \mathscr{E}_{\mathscr{P}}(N_x) + N_z N_x \mathscr{E}_{\mathscr{P}}(N_y) + N_x N_y \mathscr{E}_{\mathscr{P}}(N_z) \tag{33}$$
$$\mathscr{E}_{\mathscr{P}} \in \{\mathscr{E}_{\mathrm{FD5}}, \mathscr{E}_{\mathrm{Matr}}\}$$

because we can assume that $\mathscr{E}_{\mathscr{P}^2} = \mathscr{E}_{\mathscr{P}}$.

One can see already from comparing Eqs. (29), (30), (32), (33) that the FFT looks more expensive than matrix techniques, in particular for variable inverse mass B. In order to make the comparison more quantitative, we have chosen two typical grids as they are used in nuclear deformed Hartree–Fock or TDHF calculations: a small grid with $16 \times 16 \times 64$ points and a large grid with $32 \times 32 \times 128$ points. This gives the relative computing times as listed in Table II. It is obvious that the FFT is inferior to the full matrix scheme in both grids for the variable inverse mass B. The reason is that nuclear physics does not require too large

TABLE II

Relative computing times for One Action of the Hamiltonian in Three Cartesian Dimensions

	IBM 3090 VF			CRAY X-MP		
$N_x \times N_y \times N_z$	$\mathscr{E}_{\mathrm{FFT},B}$	$\mathscr{E}_{\mathrm{FD5},B}$	$\mathscr{E}_{\mathrm{Matr},B}$	$\mathscr{E}_{\mathrm{FFT},B}$	$\mathscr{E}_{\mathrm{FD5},B}$	$\mathscr{E}_{\mathrm{Matr},B}$
$16 \times 16 \times 64$	12.72	1.82	4.00	3.17	0.33	0.92
$32 \times 32 \times 128$	75.53	8.40	44.52	23.18	1.58	12.23
	$\mathscr{E}_{\mathrm{FFT},m}$	$\mathscr{E}_{\mathrm{FD5},m}$	$\mathscr{E}_{\mathrm{Matr},m}$	$\mathscr{E}_{\mathrm{FFT},m}$	$\mathscr{E}_{\mathrm{FD5},m}$	$\mathscr{E}_{\mathrm{Matr},m,3\mathrm{D}}$
$16 \times 16 \times 64$	3.18	0.91	2.00	0.79	0.17	0.46
$32 \times 32 \times 128$	18.88	4.20	22.26	5.80	0.79	6.11
	$\mathscr{E}_{\mathrm{FFT},\mathscr{D}}$	$\mathscr{E}_{\mathrm{FD5},\mathscr{D}}$	$\mathscr{E}_{\mathrm{Matr},\mathscr{D}}$	$\mathscr{E}_{\mathrm{FFT},\mathscr{D}}$	$\mathscr{E}_{\mathrm{FD5},\mathscr{D}}$	$\mathscr{E}_{\mathrm{Matr},\mathscr{D}}$
$16 \times 16 \times 64$	0	>0.91	2.00	0	>0.17	0.46
$32 \times 32 \times 128$	0	>4.20	22.26	0	>0.79	6.11
	$\mathscr{E}_{\mathrm{FFT},R}$	$\mathscr{E}_{\mathrm{FD5},R}$	$\mathscr{E}_{\mathrm{Matr},R}$	$\mathscr{E}_{\mathrm{FFT},R}$	$\mathscr{E}_{\mathrm{FD5},R}$	$\mathscr{E}_{\mathrm{Matr},R}$
$16 \times 16 \times 64$	6.36	1.82	4.00	1.59	0.33	0.93
$32 \times 32 \times 128$	37.76	8.40	44.52	11.59	1.58	12.23

Note. With the various cases estimated in Eqs. (29), (30), (32), and (33), using the optimised routines in each case, except for the case FD5 on the CRAY. The second index means: $B \equiv$ variable mass, Eqs. (29), (32), $m \equiv$ constant mass, Eqs. (30), (33); $\mathscr{D} \equiv$ damping, Eq. (34); $R \equiv$ quasirelativistic, Eqs. (36), (37). The index Matr stands for the action with the full matrix and FD5 for the sparse matrix of the fifth-order finite difference scheme. The last index "3D" has been omitted to keep notation short. The values from Table I have been divided by 1000 to give a proper scale for the computing times.

grids. The FFT would become advantageous for the next larger grid $64 \times 64 \times 256$. The comparison for the case of a constant mass m changes the situation. Here the FFT becomes competitive with the matrix technique.

The finite difference scheme looks favourable at first glance if one compares computing times on the same grid. But that is not a fair comparison because the finite difference scheme is much inferior in precision. We see from Fig. 1 that one needs a much smaller grid spacing, $\Delta r_{\mathrm{FD5}} \approx \frac{1}{3} \Delta r_{\mathrm{FFT}}$, in order to reach comparable precision. The finite difference scheme looks less favourable, if we compare it for the grid $32 \times 32 \times 128$ to the other schemes with the grid $16 \times 16 \times 64$ in Table II, and it is even more inferior because one should have tripled the grid for comparison and not only doubled it.

6.2.2. *Expense of Damping*

The action of the Hamiltonian is only one ingredient in static or dynamic calculations. A further step is the damping $\mathscr{D}$ which is used in the damped gradient step (7), (8) for the static Hartree–Fock iteration or for the iterative inversion (11) in the Crank–Nicholson step (10) of TDHF. The damping in cartesian 3D is given in Eq. (20). It involves only the kinetic energy and constants. Thus exact damping is possible in the momentum-space representation, and, moreover, it is a trivial multiplication by a factor which can be done at almost no cost in the spirit of the estimates above. Damping is more complicated in coordinate-space representation. It requires the separable approach (21) and each factor $\mathscr{D}_i$, $i \in \{x, y, z\}$, stands for a full matrix multiplication in one dimension. Thus the total expense of the separable damping in coordinate-space is

$$\mathscr{E}_{\mathrm{Matr}, \mathscr{D}, 3\mathrm{D}} = N_y N_z \mathscr{E}_{\mathrm{Matr}}(N_x) + N_z N_x \mathscr{E}_{\mathrm{Matr}}(N_y) + N_x N_y \mathscr{E}_{\mathrm{Matr}}(N_z). \tag{34}$$

Although the inverse of a sparse matrix is generally a full matrix, the estimate (34) is too pessimistic for the fifth-order finite difference scheme because there is a cheaper access if one solves a linear equation with the sparse matrix for $\mathscr{D}^{-1}$ each time the damping is required. Half of the Gaussian elimination, producing upper triangular form for $\mathscr{D}^{-1}$, can be done once and kept stored. The expense of the remaining second half is just about the expense of one sparse matrix multiplication if one optimizes very carefully. Thus we assume as an optimistic estimate that the expense of damping in the finite difference scheme is about the expense of one matrix multiplication and if we keep in mind that this is a lower limit; i.e., we suppose $\mathscr{E}_{\mathrm{FD5}, \mathscr{D}, 3\mathrm{D}} \approx \mathscr{E}_{\mathrm{FD5}, m, 3\mathrm{D}}$.

The resulting estimated computing times for the two grids considered are given in the third block of Table II. One step dynamic calculation consists now of one Hamiltonian action and one damping. We see from Table II that the extra expense for damping in the matrix technique and no extra cost in case of FFT leave the lead of the matrix technique for the models with variable inverse mass B almost unchanged. But it shifts the conclusion to a visible advantage of the FFT for models with constant mass m. The finite difference scheme of fifth order is again to be compared at the next larger grid to take the difference in precision into consideration. It still loses, even with the optimistic estimate of the cost of damping.

6.2.3. *Quasirelativistic Hamiltonian*

The actual expense of an action of the Hamiltonian is not only machine dependent but also model dependent, as we have seen from the comparison of the variable mass approach and the constant mass approach. There is a further possibly important variant in connection with relativistic nuclear models [4–6]. The effective Schrödinger equation for the Dirac wavefunctions couples spin-matrices and momenta in a particular way such that in the nonrelativistic limit [6]

$$h\psi = \boldsymbol{\sigma} \cdot \mathbf{p} B(\mathbf{r})\, \boldsymbol{\sigma} \cdot \mathbf{p} \psi + V(\mathbf{r})\, \psi. \tag{35}$$

The $\boldsymbol{\sigma} \cdot \mathbf{p}\psi$ can be worked out completely in momentum-space such that the momentum-space approach with FFT requires only twice two three-dimensional Fourier transformations. This yields the expense

$$\mathscr{E}_{\mathrm{FFT}, R, 3\mathrm{D}}(N_x, N_y, N_z) = 4\{N_y N_z \mathscr{E}_{\mathrm{FFT}}(N_x) + N_z N_x \mathscr{E}_{\mathrm{FFT}}(N_y) + N_x N_y \mathscr{E}_{\mathrm{FFT}}(N_z)\}. \tag{36}$$

However, in coordinate space, each term in $\boldsymbol{\sigma} \cdot \mathbf{p}\psi$ is a separate matrix multiplication. This happens twice, once before B and once after B. Altogether the expense is exactly the same as for the standard case with variable mass B, i.e.,

$$\mathscr{E}_{\mathscr{P}, R, 3\mathrm{D}}(N_x, N_y, N_z) = 2\{N_y N_z \mathscr{E}_{\mathscr{P}}(N_x) + N_z N_x \mathscr{E}_{\mathscr{P}}(N_y) + N_x N_y \mathscr{E}_{\mathscr{P}}(N_z)\}, \tag{37}$$

where again $\mathscr{E}_{\mathscr{P}} \in \{\mathscr{E}_{\mathrm{FD5}}, \mathscr{E}_{\mathrm{Matr}}\}$; see also Eq. (32).

The results for this quasirelativistic Hamiltonian (35) are given in the lowest block of Table II. We see that the momentum-space representation becomes competitive, and it is even advantagous for the larger grid if we include the cost of damping. In any case, the Hamiltonian in the form (35) has the great advantage that the spin-orbit force is already included without extra expense. One may prefer [6].

6.2.4. *Full 3D Estimate*

For the case of cartesian 3D, we have even worked out a full 3D comparison of the matrix technique with the third-order finite difference scheme which is the standard scheme in large scale applications. The matrices $\mathscr{P}_\nu$ have been evaluated from the Fourier definition of the momentum operator. We compare in Fig. 3 precision and corresponding computing time for the two schemes at various grid spacings Δr. Again we used the Hamiltonian of the form (1) with the potentials (2) in the parametrization (3). The upper part of Fig. 3 shows the precision in energy as function of the grid spacing Δr. The figure looks very similar to the spherical results in Fig. 1. The data are shifted a bit to larger Δr. This means that the one-dimensional spherical test is relevant for every other dimension. It is rather a bit more critical than the cartesian case which is not surprising because spaces with cartesian volume element are simpler. The lower part of Fig. 3 shows the corresponding computing times. These times account for the total iteration until convergence, and not just for one step. Thus they already include the effect that the iteration is somewhat slower for smaller grid spacings Δr. It is obvious that the matrix techniques are a lot more expensive at a given Δr. However, we have to compare computing times at a given precision for E. Then the matrix technique is the clear winner. For example, the precision of the third-order finite difference scheme at $\Delta r = 0.2$ fm is reached with the matrix technique at $\Delta r \approx 0.9$ fm, and the latter is orders of magnitude faster. This conclusion applies if high precision is required. If poor precision is adequate, e.g., the matrix technique at $\Delta r = 1.1$ fm yields the same accuracy as the third-order finite difference scheme at $\Delta r = 0.7$ fm. Comparison of computing times shows that both methods are equivalent at this level of precision. But remember that the latter comparison was done for an unacceptably low precision. In general, the previous conclusion persists: matrix techniques (or momentum representation) are in the end much faster for Hamiltonians with variable mass B.

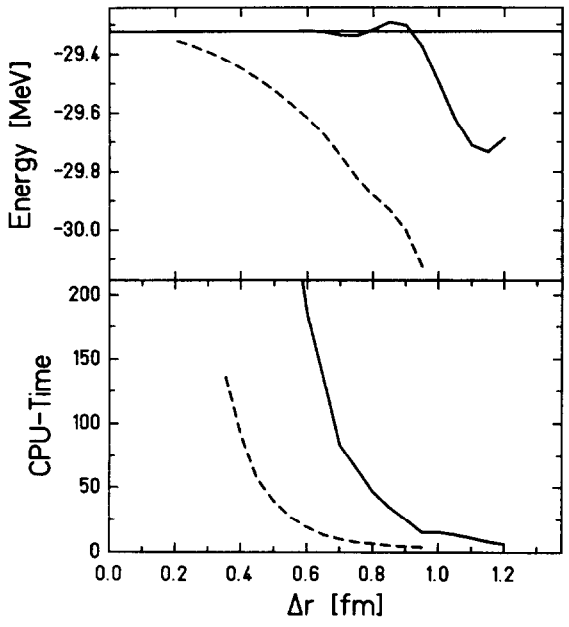

FIG. 3. Comparison of matrix technique based on a Fourier definition of the momentum (full line) with third-order finite difference scheme (dashed line). Upper part shows the energy of the 1s state as function of the grid spacing Δr. The lower part shows the corresponding CPU-time. The absolute size of the box has been kept constant at $N_x \Delta r = N_y \Delta r = N_z \Delta r = 20$ fm.

6.2.5. *Considerations on TDHF*

The estimates for the expense of time-steps in TDHF is complicated by the fact that there is a much wider choice of strategies for solution and the chosen strategy depends very much on the numerical representation of the wavefunctions.

The applications of the momentum-space techniques in TDHF [11], e.g., have used a predictor-corrector scheme for time-stepping. This scheme has the advantage that it can be performed completely within momentum-space such that the expense of one time-step is just the expense of one action of the Hamiltonian. A disadvantage is that this is an explicit scheme where the size of the time-step depends critically on the maximum energy on the grid. However, in the momentum-space representation one can easily set a cutoff in the kinetic energy and thus still use fairly large time-steps. An improvement which allows using even larger time-steps is to turn the predictor-corrector scheme into an implicit scheme by iterating the corrector. The iteration requires an inversion with the mean-field Hamiltonian which can be done also iteratively as indicated in Eq. (11). The expense of one time-step is then comparable to the expense of several static steps.

The coordinate-space techniques have usually favoured the Crank–Nicholson step with inversion in separable approximation (12). This choice is advantagous for low order finite difference schemes because the inversion could be done in each time step anew by solving a sparse linear equation. It is still an open question which scheme is most appropriate for the time-step with full matrices. The separable approach to inversion sets a rather low limit on the size of the time-step. One will very probably prefer the Crank–Nicholson step with exact inversion in full dimensionality. This inversion will again be done by iteration (11). This leads also here to the conclusion that the expense of one time-step is comparable to the expense of several static steps.

Although we see many strategies for solving the time-step in TDHF, all of them yield estimates for the expense comparable to several static steps. Thus we can take over the observations and conclusions from the static case as worked out in the previous subsections.

6.2.6. *Summary for 3D*

The above considerations have shown that matrix techniques are to be preferred for models with variable inverse mass B, momentum-space representation with FFT becomes preferable for models with constant mass m, and both techniques are competitive for the quasirelativistic model. The fifth-order finite difference scheme is inferior in all cases. It is to be kept in mind that these conclusions are drawn for nuclear physics applications which do not demand too large grids.

Occasionally there is some freedom for the decision. Then one has to take into account additional aspects. As arguments in favour of the momentum-space representation we can add that the number of relevant components for the wavefunctions can be reduced by a factor of 10. This helps in storage problems, in wave function handling (e.g., in a predictor-corrector step), and in orthonormalisation. On the other hand, we can add in favour of the matrix multiplication that the size of the grid can be handled much more flexibly and is not bound to values like 16, 32, etc. Finally, we add a comment against finite difference schemes in a 3D calculation: although the expense grows much more slowly than with both other schemes, the number of grid-points grows as N^3 and one needs a much larger grid to compete in precision. This leads to severe problems with storage of wavefunctions and with orthonormalisation in large scale problems.

6.3. 2D Axial

A momentum-space representation is disadvantagous in case of axial 2D. It has to use the Bessel transformation for which no fast scheme exists. It has to be counted like one full matrix multiplication. Then the overhead of factor two for forward and backward transformation, see the above cartesian examples, makes the momentum-space representation immediately inferior to a coordinate-space representation with full matrix multiplication. Thus we discard the momentum-space representation for the case of axial 2D.

The Hamiltonian for the case of axial 2D is given in Eq. (16). For the matrix techniques

$$h = \mathscr{P}_z B(r, z)\, \mathscr{P}_z + \frac{1}{r}\, \mathscr{P}_r (rB(r, z)\, \mathscr{P}_r) + B(r, z)\frac{m^2}{r^2} + V(r, z), \tag{38}$$

the expense for one Hamiltonian action becomes

$$\mathscr{E}_{\mathscr{P}, B, 2D} = 2(N_z \mathscr{E}_{\mathscr{P}}(N_r) + N_r \mathscr{E}_{\mathscr{P}}(N_z)), \qquad \mathscr{E}_{\mathscr{P}} \in \{\mathscr{E}_{\text{Matr}}, \mathscr{E}_{\text{FD5}}\}, \tag{39}$$

TABLE III

Relative Computing Times for One Action of the Hamiltonian in Two Axial Dimensions with the Various Cases Estimated in Eqs. (39) and (40)

	IBM 3090 VF		CRAY X-MP	
$N_x \times N_r$	$\mathscr{E}_{\text{FD5}, B}$	$\mathscr{E}_{\text{Matr}, B}$	$\mathscr{E}_{\text{FD5}, B}$	$\mathscr{E}_{\text{Matr}, B}$
16× 64	0.66	1.70	0.12	0.44
32× 128	1.54	10.90	0.28	3.10
	$\mathscr{E}_{\text{FD5}, m}$	$\mathscr{E}_{\text{Matr}, m}$	$\mathscr{E}_{\text{FD5}, m}$	$\mathscr{E}_{\text{Matr}, m}$
16× 64	0.33	0.85	0.06	0.22
32× 128	0.77	5.45	0.14	1.55
	$\mathscr{E}_{\text{FD5}, \mathscr{D}}$	$\mathscr{E}_{\text{Matr}, \mathscr{D}}$	$\mathscr{E}_{\text{FD5}, \mathscr{D}}$	$\mathscr{E}_{\text{Matr}, \mathscr{D}}$
16× 64	>0.33	0.85	>0.06	0.22
32× 128	>0.77	5.45	>0.14	1.55

Note. Using the optimised routines in each case except for the case FD5 on the CRAY. The lowest block contains the relative computing times for the damping. The index Matr stands for the action with the full matrix and FD5 for the sparse matrix of the fifth-order finite difference scheme. The last index 2D has been omitted in order to keep the notation short. The values from Table I have been divided by 100 to give short numbers for the computing times.

and for constant mass m reduces to

$$\mathscr{E}_{\mathscr{P}, m, 2D} = N_z \mathscr{E}_{\mathscr{P}}(N_r) + N_r \mathscr{E}_{\mathscr{P}}(N_z), \qquad \mathscr{E}_{\mathscr{P}} \in \{\mathscr{E}_{\text{Matr}}, \mathscr{E}_{\text{FD5}}\}. \tag{40}$$

The damping is performed in the separable approach (18). This is as expensive as the action of the Hamiltonian with constant mass, $\mathscr{E}_{\mathscr{P}, \mathscr{D}, 2D} = \mathscr{E}_{\mathscr{P}, m, 2D}$, where the estimate of Subsection 6.2.2 was used.

Similar to the previous subsections, we evaluate the estimated relative computing times for two typical grids, a small one with $N_r \times N_z = 16 \times 64$ and a large one with $N_r \times N_z = 32 \times 128$. The results in Table III show that the matrix technique is to be preferred, because the finite difference scheme provides only comparable expenses at a doubled grid, but we need a tripled grid to achieve the same precision as with full matrix technique. Further arguments in favour of the matrix technique are: a large reduction in storage-space for the wavefunctions, which also speeds up the orthonormalization in the static case, and easier vectorization.

6.4. 1D Cartesian and 1D Spherical

The Fourier transform for a one-dimensional wavefunction requires just $\mathscr{E}(N_x)$ operations. The Hamiltonian is

given as in Eq. (28), or Eq. (31), respectively, with the y- and z-terms missing. Similar reasoning as before yields

$$\begin{aligned} \mathscr{E}_{\mathrm{FFT},B,1\mathrm{D}} &= 4\mathscr{E}(N_x), & \mathscr{E}_{\mathscr{P},B,1\mathrm{D}} &= 2\mathscr{E}_{\mathscr{P}} \\ \mathscr{E}_{\mathrm{FFT},m,1\mathrm{D}} &= 2\mathscr{E}(N_x), & \mathscr{E}_{\mathscr{P},m,1\mathrm{D}} &= \mathscr{E}_{\mathscr{P}} \\ \mathscr{E}_{\mathrm{FFT},\mathscr{Q},1\mathrm{D}} &= 0, & \mathscr{E}_{\mathscr{P},\mathscr{Q},1\mathrm{D}} &= \mathscr{E}_{\mathscr{P}}, \end{aligned} \tag{41}$$

where $\mathscr{P}$ stands for $\mathscr{P} \in \{\mathrm{FD5}, \mathrm{Matr}\}$.

The case of spherical 1D combines arguments from axial 2D and cartesian 1D. The momentum-space representation is disadvantagous for this case because it employs the Fourier–Bessel transformation. But this is a full matrix operation and the overhead of forward and backward transformations makes this scheme inferior. It remains to compare matrix techniques with fifth-order finite difference scheme. This comparison can directly be taken over from Eq. (41).

The estimated expenses are all trivial factors of the raw expenses given in Table I and can be read off from there using Eq. (41). Note that the finite difference schemes (FD5) become clearly advantagous in one dimension. The expenses for the four times larger grid are still lower than the expenses on the small grid in both other methods. The storage problem which is a big hindrance for finite difference schemes in 3D is uncritical in 1D. In addition, there are particularly efficient techniques for the inversion of the fifth-order kinetic energy in one dimension [24, 25]. And furthermore, round-off errors are not so dramatic in one dimension such that one may even consider higher order finite difference schemes. Altogether, finite difference schemes are the method of choice in one-dimensional problems. And that is exactly what has been used successfully for a long time for this type of calculation [25].

7. CONCLUSION

We have studied standard gridding techniques for solving the Schrödinger equation using a representation in coordinate- or momentum-space. The central task in all schemes is to provide a reliable description of the momentum operators and subsequently of the kinetic energy. The momentum-space technique combines with the fast Fourier transformation (FFT) for fast switching from momentum-space (where the kinetic energy is evaluated) to coordinate-space (where local potentials are evaluated). The coordinate-space techniques represent momenta by matrices. We consider finite difference schemes which generate sparse momentum-matrices, and B-splines or Fourier definition which generate fully packed momentum-matrices.

We have compared these schemes with respect to precision and computing times. The comparison of computing times was done in a mixed approach: computing times for the basic operations in one spatial dimension have been evaluated on two existing vector machines, and IBM 3090 VF and a CRAY X/MP, and from these numbers, estimates for the complete operations (action of the Hamiltonian, damping) have been worked out analytically in many different dimensions. This procedure gives interesting insights and provides valuable indications for a decision on the optimal schemes. Of course, this method misses the maximum possible speed which may be reached by laborious optimization of a code for each of the various techniques. We assume that possible gains in speed are about equal in each technique such that our procedure gives reliable estimates of the expected relative computing times.

The comparison of the precision as a function of the grid-spacing has shown that both, the Fourier definition of the momentum as well as the B-splines representation, provide almost the same precision and that both are very precise schemes, superior to the finite difference schemes. Both winning schemes have reached the point where the precision of the kinetic energy is very accurate and where the local treatment of the potential sets the limits. One may think of improving the precision further by nonlocal handling of the potentials. This remains an open question.

The comparison of computing times depends on the form of the Hamiltonian. We have distinguished three cases, variable mass, constant mass, and quasirelativistic Hamiltonian. The conclusions differ for the three cases. The conclusions also depend, of course, on the size of the grid and of the dimensionality of the problem. In all our comparisons we have considered grid sizes of $N = 16, \ldots, 128$ as they occur typically in nuclear physics. There are many detailed pro's and con's in the various cases. We try to give here a condensed record of results:

Cartesian 3D. The coordinate-space representation with full matrices for the momenta is to be preferred for variable mass B; the momentum-space representation with FFT becomes advantageous for constant mass m; both methods are comparable for the quasirelativistic Hamiltonian. The finite difference scheme is to be discarded in every case.

Axial 2D. The coordinate-space representation with full matrices for the momenta is to be preferred in any case.

Cartesian and spherical 1D. Finite difference schemes of at least fifth order are the preferred technique in any case.

ACKNOWLEDGMENT

The authors thank the computing center of the GSI and the Höchstleistungsrechenzentrum of the KFA Jülich for providing the computing time for these extensive tests.

REFERENCES

1. P. Quentin and H. Flocard, *Annu. Rev. Nucl. Part. Sci.* **28**, 523 (1978).
2. K. Goeke and P.-G. Reinhard, *Lecture Notes in Physics*, Vol. 171 (Springer-Verlag, New York/Berlin, 1982).
3. A. S. Umar, M. R. Strayer, P.-G. Reinhard, K. T. R. Davies, and S.-J. Lee, *Phys. Rev. C* **40**, 706 (1989).
4. B. D. Serot and J. D. Walecka, *Adv. Nucl. Phys.* **16**, 1 (1986).
5. L. S. Celenza and C. M. Shakin, *Relativistic Nuclear Physics—Theories of Structure and Scattering* (World Scientific, Singapore, 1986).
6. P.-G. Reinhard, *Rep. Prog. Phys.* **52**, 439 (1989).
7. P. Hoodbhoy and J. W. Negele, *Nucl. Phys. A* **288**, 23 (1977).
8. K. T. R. Davies and S. E. Koonin, *Phys. Rev. C* **23**, 2042 (1981).
9. S.-J. Lee, J. Fink, A. B. Balantekin, M. R. Strayer, A. S. Umar, P.-G. Reinhard, J. A. Maruhn, and W. Greiner, *Phys. Rev. Lett.* **57**, 2916 (1986); **59**, 1171 (1987); **60**, 163 (1988).
10. J. Fink, Ph.D. thesis, University of Frankfurt, 1988.
11. H. Stöcker, R. Y. Cusson, J. Maruhn, and W. Greiner, *Z. Phys. A* **294**, 125 (1980).
12. C. Bottcher and M. R. Strayer, *Ann. Phys. (N.Y.)* **175**, 64 (1987).
13. J. J. Bay, R. Y. Cusson, J. Wu, P.-G. Reinhard, H. Stöcker, W. Greiner, and M. R. Strayer, *Z. Phys. A* **326**, 269 (1987).
14. C. de Boor, *A Practical Guide to Splines* (Springer-Verlag, New York, 1978).
15. A. S. Umar and V. Oberacker, private communication, 1990.
16. O. Gunnarson and B. I. Lundquist, *Phys. Rev. B* **13**, 4274 (1976).
17. P.-G. Reinhard, J. Friedrich, and N. Voegeler, *Z. Phys. A* **316**, 207 (1984).
18. P. Bonche, S. Koonin, and J. W. Negele, *Phys. Rev. C* **13**, 1226 (1976).
19. P.-G. Reinhard, M. Brack, and O. Genzken, *Phys. Rev. A* **41**, 5568 (1990).
20. P.-G. Reinhard and R. Y. Cusson, *Nucl. Phys. A* **378**, 418 (1982).
21. R. Y. Cusson, P.-G. Reinhard, M. R. Strayer, J. A. Maruhn, and W. Greiner, *Z. Phys. A* **320**, 475 (1985).
22. H. Flocard, S. Koonin, and M. S. Weiss, *Phys. Rev. C* **17**, 1682 (1978).
23. M. Abramowitz and I. A. Stegun, *Handbook of Mathematical Functions* (Dover, New York, 1972).
24. R. Zurmühl, *Praktische Mathematik*, Chap. VII, Sect. 3 (Springer-Verlag, Berlin, 1963).
25. P.-G. Reinhard, in *Numerical Physics*, edited by S. Koonin, J. A. Maruhn, and M. R. Zirnbauer (Springer-Verlag, Heidelberg, 1990).