Authors: S.I. Vinitsky, A.A. Gusev and O. Chuluunbaatar

Language: Fortran

The subroutine TIME6T 2.0 calculates numerical solutions $\Psi(\mathbf{r},t)$ of the Cauchy problem for the time-dependent Schrödinger equation in d-dimensional space

$$\imath\frac{\partial\Psi(\mathbf{r},t)}{\partial t}=\left(-\frac{1}{2}\nabla^2_{\mathbf{r}}+U(\mathbf{r})+f(\mathbf{r},t)\right)\Psi(\mathbf{r},t),\quad \Psi(\mathbf{r},t_0)=\Psi_0(\mathbf{r}), \tag{1}$$

up to $2M$ ($M=1,2,3$) order of accuracy in the time step τ of a uniform grid $\Omega_\tau[t_0,T]=\{t_0,t_{k+1}=t_k+\tau,(k=0,1,\ldots,K-1),t_K=T\}$ covering the time interval $t\in[t_0,T]$, in the frame of the Kantorovich method [1] using the explicit Magnus expansion [2] of the evolution operator with the additional gauge transformations and their Padé approximations.

We require the solution $\Psi(\mathbf{r},t)$ to be continuous and to have general first derivatives integrable with square, and to belong to the Sobolev space $\Psi(\mathbf{r},t)\in\mathbf{W}^1_2(\mathbf{R}^d\otimes[t_0,T])$ ($\Psi_0(\mathbf{r})\in\mathbf{W}^1_2(\mathbf{R}^d)$). Also, we suppose that the functions $U(\mathbf{r})$ and $f(\mathbf{r},t)$ are functions of high smoothness by the spatial variable $\mathbf{r}$, and the function $f(\mathbf{r},t)$ has partial derivatives to the order of $2M$ to be continuous by time variables t. The normalization condition reads

$$\|\Psi(\mathbf{r},t)\|^2=\int|\Psi(\mathbf{r},t)|^2d\mathbf{r}=1,\quad t\in[t_0,T]. \tag{2}$$

Remark. At $M=3$ subroutine TIME6T 2.0 is realized only for potentials $f(\mathbf{r},t)$ satisfying the conditions $\nabla^2_{\mathbf{r}}\partial_t f(\mathbf{r},t)\equiv 0$.

In the Kantorovich method, the solution $\Psi(\mathbf{r},t)$ is expanded over the oneparameter basis functions $\{B_j(\Omega;r)\}_{j=1}^{j_{\max}}$:

$$\Psi(\mathbf{r},t)=\sum_{j=1}^{j_{\max}}B_j(\Omega;r)\chi_j(r,t). \tag{3}$$

In equation (3) the vector function $\boldsymbol{\chi}(r,t)=(\chi_1(r,t),\ldots,\chi_{j_{\max}}(r,t))^T$ is unknown, and the components of the surface vector function $\mathbf{B}(\Omega;r)=(B_1(\Omega;r),\ldots,B_{j_{\max}}(\Omega;r))^T$ form an orthonormal basis with respect to the set of angular coordinates Ω for each value of hyperradius r, which is treated here as a given parameter. The basis functions $B_j(\Omega;r)\in\mathcal{F}_r\sim\mathbf{L}_2(S^{d-1}(\Omega))$ are determined as solutions of the following parametric eigenvalue problem:

$$\left(-\frac{1}{2r^2}\hat{\mathbf{\Lambda}}^2_\Omega+U(\mathbf{r})\right)B_j(\Omega;r)=E_j(r)B_j(\Omega;r), \tag{4}$$

where $\hat{\mathbf{\Lambda}}^2_\Omega$ is the generalized self-adjoint angular momentum operator. The eigenfunctions of this problem satisfy the same boundary conditions in angular variable Ω for $\Psi(\mathbf{r},t)$ at each fixed value of the parameter r and are normalized as

$$\Big\langle B_i(\Omega;r)\Big|B_j(\Omega;r)\Big\rangle_\Omega=\int\overline{B}_i(\Omega;r)B_j(\Omega;r)d\Omega=\delta_{ij}. \tag{5}$$

An averaging of Eq. (1) by the basis $B_j(\Omega;r)$, leads to a system of $j_{\max}$ ordinary second-order differential equations for the $\boldsymbol{\chi}(r,t)$:

$$\imath\,\mathbf{I}\frac{\partial\boldsymbol{\chi}(r,t)}{\partial t}=\mathbf{H}(r,t)\boldsymbol{\chi}(r,t),\quad \boldsymbol{\chi}(r,t_0)=\boldsymbol{\chi}_0(r), \tag{6}$$

with

$$\mathbf{H}(r,t)=-\frac{1}{2r^{d-1}}\mathbf{I}\frac{\partial}{\partial r}r^{d-1}\frac{\partial}{\partial r}+\mathbf{V}(r)+\tilde{\mathbf{Z}}^{(1)}(r,t)+\mathbf{Q}(r)\frac{\partial}{\partial r}+\frac{1}{r^{d-1}}\frac{\partial r^{d-1}\mathbf{Q}(r)}{\partial r}. \tag{7}$$

Here $\mathbf{I}$, $\mathbf{V}(r)$, $\tilde{\mathbf{Z}}^{(1)}(r,t)$ and $\mathbf{Q}(r)$ are matrices of dimension $j_{\max}\times j_{\max}$, whose elements are given by the relation

$$
\begin{aligned}
V_{ij}(r) &= V_{ji}(r) = \frac{E_i(r)+E_j(r)}{2}\delta_{ij} + \frac{1}{2}\left\langle \frac{\partial B_i(\Omega;r)}{\partial r}\middle| \frac{\partial B_j(\Omega;r)}{\partial r}\right\rangle_{\Omega},\\
Q_{ij}(r) &= -Q_{ji}(r) = -\frac{1}{2}\left\langle B_i(\Omega;r)\middle| \frac{\partial B_j(\Omega;r)}{\partial r}\right\rangle_{\Omega},\\
\tilde{Z}^{(1)}_{ij}(r,t) &= \tilde{Z}^{(1)}_{ji}(r,t) = \left\langle B_i(\Omega;r)\middle| f(\mathbf{r},t)\middle| B_j(\Omega;r)\right\rangle_{\Omega}.
\end{aligned}
\tag{8}
$$

In solving the numerical solution of the problem (6)–(8) at each $\boldsymbol{\chi}(r,t_k)$ transforming into $\boldsymbol{\chi}(r,t_{k+1})$ we use the following implicit numerical scheme [2] with the symmetric operators $\tilde{\mathbf{A}}^{(M)}_k$, $\tilde{\mathbf{S}}^{(M)}_k$:

$$
\begin{aligned}
&\tilde{\boldsymbol{\chi}}^0_k = \boldsymbol{\chi}(r,t_k),\\
&\left(\mathbf{I} - \frac{\overline{\alpha}^{(L)}_\eta \tilde{\mathbf{S}}^{(M)}_k}{2L}\right)\tilde{\boldsymbol{\chi}}^{\eta/L}_k = \left(\mathbf{I} - \frac{\alpha^{(L)}_\eta \tilde{\mathbf{S}}^{(M)}_k}{2L}\right)\tilde{\boldsymbol{\chi}}^{(\eta-1)/L}_k,\ \eta = 1,\dots,L,\\
&\hat{\boldsymbol{\chi}}^0_k = \tilde{\boldsymbol{\chi}}^1_k,\\
&\left(\mathbf{I} + \frac{\tau\overline{\alpha}^{(M)}_\zeta \tilde{\mathbf{A}}^{(M)}_k}{2M}\right)\hat{\boldsymbol{\chi}}^{\zeta/M}_k = \left(\mathbf{I} + \frac{\tau\alpha^{(M)}_\zeta \tilde{\mathbf{A}}^{(M)}_k}{2M}\right)\hat{\boldsymbol{\chi}}^{(\zeta-1)/M}_k,\ \zeta = 1,\dots,M,\\
&\tilde{\boldsymbol{\chi}}^0_k = \hat{\boldsymbol{\chi}}^1_k,\\
&\left(\mathbf{I} + \frac{\overline{\alpha}^{(L)}_\eta \tilde{\mathbf{S}}^{(M)}_k}{2L}\right)\tilde{\boldsymbol{\chi}}^{\eta/L}_k = \left(\mathbf{I} + \frac{\alpha^{(L)}_\eta \tilde{\mathbf{S}}^{(M)}_k}{2L}\right)\tilde{\boldsymbol{\chi}}^{(\eta-1)/L}_k,\ \eta = 1,\dots,L,\\
&\boldsymbol{\chi}(r,t_{k+1}) = \tilde{\boldsymbol{\chi}}^1_k.
\end{aligned}
\tag{9}
$$

Here $L = [\frac{M}{2}]$, I is the unit operator. The coefficients, $\alpha^{(M)}_\zeta$ ($\zeta = 1,\dots,M$, $M \geq 1$), stand for the roots of the polynomial equation ${}_1F_1(-M,-2M,2M\imath/\alpha) = 0$, where ${}_1F_1(a,b,x)$ is the confluent hypergeometric function and the overline indicates the complex conjugate. At $M = 1,2,3$ the operators $\tilde{\mathbf{A}}^{(M)}_k$, $\tilde{\mathbf{S}}^{(M)}_k$ are given by the relations

$$
\begin{aligned}
\tilde{\mathbf{A}}^{(1)}_k &= \mathbf{H}(r,t_c), & \tilde{\mathbf{S}}^{(1)}_k &= \mathbf{0},\\
\tilde{\mathbf{A}}^{(2)}_k &= \tilde{\mathbf{A}}^{(1)}_k + \tilde{\mathbf{G}}^{(2)}, & \tilde{\mathbf{S}}^{(2)}_k &= \tilde{\mathbf{S}}^{(1)}_k + \tilde{\mathbf{Z}}^{(2)},\\
\tilde{\mathbf{A}}^{(3)}_k &= \tilde{\mathbf{A}}^{(2)}_k + \tilde{\mathbf{G}}^{(3)}, & \tilde{\mathbf{S}}^{(3)}_k &= \tilde{\mathbf{S}}^{(2)}_k + \tilde{\mathbf{Z}}^{(3)},
\end{aligned}
\tag{10}
$$

$$
\begin{aligned}
\tilde{G}^{(2)}_{ij}(r,t_c) &= \tau^2\left\langle B_i(\Omega;r)\middle|\frac{1}{24}\,\ddot{f}\middle| B_j(\Omega;r)\right\rangle_{\Omega},\\
\tilde{Z}^{(2)}_{ij}(r,t_c) &= \tau^2\left\langle B_i(\Omega;r)\middle|\frac{1}{12}\,\dot{f}\middle| B_j(\Omega;r)\right\rangle_{\Omega},\\
\tilde{G}^{(3)}_{ij}(r,t_c) &= \tau^4\left\langle B_i(\Omega;r)\middle|\frac{1}{1920}\,\dddot{f} + \frac{1}{1440}\left(\nabla_{\mathbf{r}}\,\dot{f}\right)^2 - \frac{1}{720}\left(\nabla_{\mathbf{r}}\,\ddot{f}\right)\left(\nabla_{\mathbf{r}}(U+f)\right)\middle| B_j(\Omega;r)\right\rangle_{\Omega},\\
\tilde{Z}^{(3)}_{ij}(r,t_c) &= \tau^4\left\langle B_i(\Omega;r)\middle|\frac{1}{480}\,\dddot{f} + \frac{1}{720}\left(\nabla_{\mathbf{r}}\,\dot{f}\right)\left(\nabla_{\mathbf{r}}(U+f)\right)\middle| B_j(\Omega;r)\right\rangle_{\Omega},
\end{aligned}
\tag{11}
$$

where $f \equiv f(x,t_c)$, $\dot{f} \equiv \partial_t f(x,t)|_{t=t_c}$, ..., $U \equiv U(\mathbf{r})$ and $t_c = t_k + \tau/2$.

Methods:

The considered scheme was constructed on the basis of the algorithms published in [3]. Further discretization of the resulting problem (9)–(11) at each layer $t = t_k$ is implemented by means of the finite-element method [4] by spatial variable $r \in [0, r_{\max}]$, using Lagrange interpolation polynomials up to the order $p = 8$ at suitable smoothness of the solution.

Test:

Test example is carried out from paper [5] for an exact solvable two-dimensional model (1), (2) with potentials $U(\mathbf{r}) = \omega^2 r^2/8$ and $f(\mathbf{r},t) = r(f_1(t)\cos(\theta) + f_2(t)\sin(\theta))$, where $f_1(t) = \cos(\omega t/2)E_1(t) - \sin(\omega t/2)E_2(t)$, $f_2(t) = -\sin(\omega t/2)E_1(t) - \cos(\omega t/2)E_2(t)$, $E_j(t) = a_j \sin(\omega_j t)$. In test run in interval $t \in [0, 0.4]$ we use the following values of parameters: $\omega = 4\pi$, $\omega_1 = \pi$,$\omega_2 = 2\pi$, $a_1 = 1$, $a_2 = 2$.

References:

[1] Kantorovich L.V. and Krylov V.I. *Approximate Methods of Higher Analysis*. New York, Wiley, 1964.

[2] Magnus W. Commun. Pure Appl. Math., 1954, v. 7, pp. 649–673.

[3] Chuluunbaatar O. et al. J. Phys. A, 2008, v. 41, pp. 295203–1–25.

[4] Bathe K.J. *Finite element procedures in engineering analysis*. Englewood Cliffs, Prentice Hall, New York, 1982.

[5] Chuluunbaatar O. et al. Phys. Rev E, 2008, v. 78, pp. 017702–1–4.


```
Structure:

      SUBROUTINE
      Name: TIME6T
      Internal subroutines: TMSOLV, ASSMBN, ASSMBT, EMASSD, ESTIFD, ESTIFN,
                            ESTITD, ESTITN, ADDVEC, HQPOTN, SGPOTN, BOUNDC,
                            COLMHT, ERRDIM, FEGRID, MAXHT, GAULEG, NODGEN,
                            SHAPEF, MULTC, MULTCC, REDBAC, DECOMC
      External subroutines: POTCAL, POTTIM, DINIT (user-supplied subroutines)

Usage:

      CALL TIME6T(TITLE,MDIM,IDIM,NPOL,TMIN,TMAX,TAU,ITORDR,
     1            IPRINT,IPRSTP,IPRITT,NMESH,RMESH,IBOUND,FNOUT,
     2            IOUT,POTEN,IOUP,POTTM,IOUD,FMATR,IOUM,EVWFN,
     3            IOUF,TOT,ITOT,ZTOT,MTOT,MITOT,MZTOT)
      INPUT:      TITLE,MDIM,IDIM,NPOL,TMIN,TMAX,TAU,ITORDR,
                  IPRINT,IPRSTP,IPRITT,NMESH,RMESH,IBOUND,FNOUT,
                  IOUT,POTEN,IOUP,POTTM,IOUD,FMATR,IOUM,EVWFN,
                  IOUF,TOT,ITOT,ZTOT,MTOT,MITOT,MZTOT:

 TITLE   -- character symbol, title of the run to be printed on the output
            listing.
 MDIM    -- integer number, number of coupled differential equations.
 IDIM    -- integer number, dimension of the envelope space.
 NPOL    -- integer number, order of interpolating Lagrange polynomials.
 TMIN    -- double precision number, begin of time interval.
 TMAX    -- double precision number, end of time interval.
 TAU     -- double precision number, time step of time variable t.
 ITORDR  -- integer number, order of Magnus expansion, 1 , 2 , 3.
 IPRINT  -- integer number, level of print, 0 , 1 , 2.
 IPRSTP  -- integer number, step of spatial variable r with which solutions
            are printed out.
 IPRITT  -- integer number, step of time variable t with which solutions
            are printed out.
 NMESH   -- integer number, dimension of array RMESH. NMESH always should be
            odd and > = 3.
 RMESH   -- double precision array, array RMESH contains information about
            subdivision of interval [0,r_max] of space variable x on
            subintervals. The whole interval [0,r_max] is divided as
            follows: RMESH(1) = 0, RMESH(NMESH) = r_max, and the values
```

```
            of RMESH(I) set the number of elements for each subinterval
            [RMESH(I-1), RMESH(I+1)], where I=2, 4, ... , NMESH-1.
 IBOUND  -- integer number, parameter defining the type of boundary
            conditions set in the boundary points r = 0 and r = r_max:
            = 1 -- the Dirichlet -- Dirichlet boundary conditions;
            = 2 -- the Dirichlet -- Neumann boundary conditions;
            = 3 -- the Neumann -- Dirichlet boundary conditions;
            = 4 -- the Neumann -- Neumann boundary conditions.
 FNOUT   -- character symbol, name of the output file for printing out
            the results of the calculation.
 IOUT    -- integer number, number of the output logical device for printing
            out the results of the calculation.
 POTEN   -- character symbol, name of the scratch file for potential matrices.
 IOUP    -- integer number, number of the logical device POTEN.
 POTTM   -- character symbol, name of the scratch file for potential matrices.
 IOUD    -- integer number, number of the logical device POTTM.
 FMATR   -- character symbol, name of the scratch file for storing matrices.
 IOUM    -- integer number, number of the logical device FMATR.
 EVWFN   -- character symbol, name of the output file for storing the results
            of the calculation, namely, finite-element grid points, and
            solutions. It is used only if IOUF > 0.
 IOUF    -- integer number, number of the logical device EVWFN.
 TOT     -- double precision array, working vector.
 ITOT    -- integer array, working vector.
 ZTOT    -- double complex array, working vector.
 MTOT    -- integer number, dimension of the working array TOT.
 MITOT   -- integer number, dimension of the working array ITOT.
 MZTOT   -- integer number, dimension of the working array ZTOT.

     OUTPUT:
            WRITE(IOUF) NN,NGRID,TT,TAU,(XGRID(J),J=1,NGRID),
    1                   (ZU(J),J=1,NN)

 NGRID -- integer number, the number of finite-element grid points.
 NN    -- integer number, the number of finite-element grid points of solution.
 TT    -- double precision number, the value of time variable t.
 TAU   -- double precision number, the time step of time variable t.
 XGRID -- double precision array, contains the values of the finite-element grid points.
 ZU    -- double complex array, contains calculated solution.
```

POTCAL is the name of the user-supplied subroutine which calculates the potential matrices V and Q and should be written as follows:

```
      SUBROUTINE POTCAL(RHO,VV,QQ,MDIM,IOUT)
      IMPLICIT REAL*8 (A-H,O-Z)
      DIMENSION VV(MDIM,MDIM),QQ(MDIM,MDIM)
      RETURN
      END
```

POTTIM is the name of the user-supplied subroutine which calculates the potential matrices G and Z and should be written as follows:

```
      SUBROUTINE POTTIM(RHO,GG,ZZ,TT,TAU,ITORDR,MDIM,IOUT)
      IMPLICIT REAL*8 (A-H,O-Z)
      DIMENSION GG(MDIM,MDIM),ZZ(MDIM,MDIM)
      RETURN
```

```
      END
```

DINIT is the name of the user-supplied subroutine which calculates the initial function and should be written as follows:

```
      SUBROUTINE DINIT(KEY,MDIM,NN,NGRID,XGRID,ZU)
      IMPLICIT REAL*8 (A-H,O-Y)
      IMPLICIT COMPLEX*16 (Z)
      DIMENSION ZU(NN),XGRID(NGRID)
      RETURN
      END
```

```
 RHO   -- double precision number, the value of spatial variable r.
 TT    -- double precision number, the value of time variable t.
 TAU   -- double precision number, the time step of time variable t.
 KEY   -- integer number, If IBOUND >= 3 then KEY = 0, else KEY = 1.
```

Remarks:

1. In subroutine TIME6T 2.0 is used the dynamic distribution storages as one-dimensional arrays TOT, ZTOT, ITOT, RMESH. In test example their dimension is set by the values MTOT=2 000 000, MZTOT=1 000 000, MITOT=300 000, NMESH1=5. If user has set insufficient values of dimension above arrays the message about an error is printed and the execution of the program is aborted. In the last case, in order to carry out the required calculation it is necessary to increase the dimensions of arrays to the quantity taken from the message.

2. For the accuracy control over the numerical solution by step TAU, the Runge coefficient on four twice condensed grids is additionally calculated, using the subroutine RUNGE.