Fast Fourier transforms in CPMD

Jürg Hutter
MPI für Festkörperforschung
Stuttgart

May 23, 1995

Fast Fourier Transform (FFT)

Fourier transform pair

$$\begin{aligned} F(\omega) &= \int_{-\infty}^{\infty} f(t) e^{i\omega t} dt \\ f(t) &= \frac{1}{2\pi} \int_{-\infty}^{\infty} F(\omega) e^{-i\omega t} d\omega \end{aligned}$$

In our case t is the direct space and ω is reciprocal space (G - space).

Properties of Fourier transforms

if	then
f(t) is real	$F(-\omega) = [F(\omega)]^*$
f(t) is imaginary	$F(-\omega) = -[F(\omega)]^*$
f(t) is even	$F(-\omega) = F(\omega)$
f(t) is odd	$F(-\omega) = -F(\omega)$
f(t) is real and even	$F(\omega)$ is real and even
f(t) is real and odd	$F(\omega)$ is imaginary and odd
f(t) is imaginary and even	$F(\omega)$ is imaginary and even
f(t) is imaginary and odd	$F(\omega)$ is real and odd

$$\begin{aligned} f(at) &\iff \frac{1}{|a|} F\left(\frac{\omega}{a}\right) \\ \frac{1}{|b|} f\left(\frac{t}{b}\right) &\iff F(b\omega) \\ f(t - t_o) &\iff F(\omega) e^{i\omega t_o} \\ f(t) e^{-i\omega_o t} &\iff F(\omega - \omega_o) \end{aligned}$$

Discrete Fourier transforms

Sampling of functions

$$\begin{aligned} f_k &= f(t_k) \\ t_k &= k\Delta \quad k = 0, 1, 2, \ldots, N-1 \\ F_n &= F(\omega_n)/\Delta \\ \omega_n &= 2\pi \frac{n}{N\Delta} \quad n = 0, 1, 2, \ldots, N-1 \end{aligned}$$

$$\begin{aligned} F(\omega_n) &= \int_{-\infty}^{\infty} f(t) e^{i\omega_n t} dt \\ &\approx \sum_{k=0}^{N-1} f_k e^{i\omega_n t_k} \Delta \\ &= \Delta \sum_{k=0}^{N-1} f_k e^{ikn(2\pi/N)} \end{aligned}$$

Inverse transform

$$f_k = \frac{1}{N} \sum_{n=0}^{N-1} F_n e^{-ikn(2\pi/N)}$$

The formula for a discrete Fourier transform can be rewritten in the form

$$F_n = \sum_{k=0}^{N-1} W^{nk} f_k$$

where $W = \exp(i2\pi/N)$.

$$\begin{aligned} F_n &= \sum_{k=0}^{N-1} e^{2\pi ink/N} f_k \\ &= \sum_{k=0}^{N/2-1} e^{2\pi in(2k)/N} f_{2k} + \sum_{k=0}^{N/2-1} e^{2\pi in(2k+1)/N} f_{2k+1} \\ &= \sum_{k=0}^{N/2-1} e^{2\pi ink/(N/2)} f_{2k} + \sum_{k=0}^{N/2-1} e^{2\pi ink/(N/2)} f_{2k+1} e^{2\pi in/N} \\ &= F_n^e + W^n F_n^o \end{aligned}$$

This factorization can be proceeded until the length of the final transform is 1. This procedure reduces the cost of a discrete Fourier transform from N^2 to N log(N).

3-Dimensional FFT

$$F(n_1,n_2,n_3) = \sum_{k_3=0}^{N_3-1}\sum_{k_2=0}^{N_2-1}\sum_{k_1=0}^{N_1-1} e^{i2\pi k_1 n_1/N_1} e^{i2\pi k_2 n_2/N_2} e^{i2\pi k_3 n_3/N_3} f(k_1,k_2,k_3)$$

The order of the operations is interchangeable.

- $N_2 N_3$ 1-D FFT of length N_1
- $N_1 N_3$ 1-D FFT of length N_2
- $N_1 N_2$ 1-D FFT of length N_3

The total number of operations is proportional to $3N^2 \log(N)$.

Optimal complex to real transforms

We want to make use of the special structure of our wavefunctions. f(t) is real $\Rightarrow$ F(ω) = F(-ω)*.

Real to complex transform

Define a new function

$$g(t) = f_1(t) + i f_2(t)$$

then we get for the transformed function

$$\begin{aligned} G(\omega) &= F_1(\omega) + iF_2(\omega) \\ G(-\omega) &= F_1(-\omega) + iF_2(-\omega) \\ &= F_1(\omega)^* + iF_2(\omega)^* \end{aligned}$$

We can calculate the two new functions $G(\omega)+G(-\omega)$ and $G(\omega)-G(-\omega)$.

$$\begin{aligned} G(\omega)+G(-\omega) &= F_1(\omega) + F_1(\omega)^* + i\left(F_2(\omega) + F_2(\omega)^*\right) \\ &= 2\mathrm{Re}\left[F_1(\omega)\right] + 2i\mathrm{Re}\left[F_2(\omega)\right] \\ G(\omega)-G(-\omega) &= 2i\mathrm{Im}\left[F_1(\omega)\right] - 2\mathrm{Im}\left[F_2(\omega)\right] \end{aligned}$$

and we find

$$\begin{aligned} F_1(\omega) &= \frac{1}{2}\left(\mathrm{Re}\left[G(\omega)+G(-\omega)\right] + i\mathrm{Im}\left[G(\omega)-G(-\omega)\right]\right) \\ F_2(\omega) &= \frac{1}{2}\left(\mathrm{Im}\left[G(\omega)+G(-\omega)\right] + i\mathrm{Re}\left[G(\omega)-G(-\omega)\right]\right) \end{aligned}$$

complex to real transform

We define

$$G(\omega) = F_1(\omega) + iF_2(\omega)$$

then we get for the functions in direct (time) space

$$\begin{aligned} g(t) &= f_1(t) + f_2(t) \\ f_1(t) &= \mathrm{Re}\,[g(t)] \\ f_2(t) &= \mathrm{Im}\,[g(t)] \end{aligned}$$

Example: CPMD routine that multiplies wavefunctions with the local potentials.

```
C     ==================================================================
      SUBROUTINE VPSI(IA,NTRA,C0,C2,F,NSTATE,NGW,KRWFN)
C     ==--------------------------------------------------------------==
      IMPLICIT REAL*8 (A-H,O-Z)
      INCLUDE 'apar.inc'
      INCLUDE 'spar.inc'
      INCLUDE 'cppt.inc'
      INCLUDE 'cnst.inc'
      INCLUDE 'geq0.inc'
      INCLUDE 'fft.inc'
      INCLUDE 'parm.inc'
      INCLUDE 'para.inc'
      INCLUDE 'rswf.inc'
      INCLUDE 'fpoint.inc'
      LOGICAL KRWFN
      COMPLEX*16 C0(NGW,*),C2(NGW,*),PSI(LEAD,*),PSIRE(LEADR,*),FP,FM
      DIMENSION RHOE(*),F(*)
      POINTER (IP_PSIRE,PSIRE)
C     ==--------------------------------------------------------------==
      CALL TISET('      VPSI',ISUB)
      NNR1=(NR1+1)*NR2S*NR3S
#ifdef PARALLEL
      LEAD  = (NR1S+1)*NGRAYS
      LEADR = NNR1
#else
      LEAD  = NNR1
      LEADR = NNR1
```

```
#endif
      IP_PSIRE = LOC(PSI(1,1))
C     ==--------------------------------------------------------------==
C     == Initialize PSI, which will contain the wavefunctions         ==
C     ==--------------------------------------------------------------==
      IF(KRWFN) THEN
        DO IB=1,NTRA
          IWF=(IA+2*(IB-1)+1)/2
          CALL DCOPY(2*NNR1,RSWF(1,IWF),1,PSIRE(1,IB),1)
        ENDDO
      ELSE
        CALL AZZERO(PSI,2*MAXFFT*NTRA)
C     ==--------------------------------------------------------------==
C     == Store the wavefunctions in array PSI in the following way:   ==
C     == Since we use the Gamma point only, the wavefunctions in real ==
C     == space are real. This fact is used to save time by using      ==
C     == complex arrays where the wavefunctions of 2 different states ==
C     == (i and i+1) are combined, and performing complex Fourier     ==
C     == Transforms. Warning: If you use K-points other than Gamma,   ==
C     == or zone boundary, this symmetry is broken and the trick is   ==
C     == not valid anymore. First step is to build the array PSI.     ==
C     ==--------------------------------------------------------------==
        DO IB=1,NTRA
          I=IA+2*(IB-1)
          IS1 = I
          IS2 = I+1
          IF(IS2.GT.NSTATE) THEN
            DO IG=1,NGW
              PSI(NZH(IG),IB)=C0(IG,IS1)
              PSI(INDZ(IG),IB)=DCONJG(C0(IG,IS1))
            ENDDO
            IF(GEQ0) PSI(NZH(1),IB)=C0(1,IS1)
          ELSE
            DO IG=1,NGW
              PSI(NZH(IG),IB)=C0(IG,IS1)+UIMAG*C0(IG,IS2)
              PSI(INDZ(IG),IB)=DCONJG(C0(IG,IS1))+
     *                                    UIMAG*DCONJG(C0(IG,IS2))
            ENDDO
            IF(GEQ0) PSI(NZH(1),IB)=C0(1,IS1)+UIMAG*C0(1,IS2)
          ENDIF
        ENDDO
C     ==--------------------------------------------------------------==
C     == Transform the wavefunction to real space                     ==
C     ==--------------------------------------------------------------==
```

```
        CALL INVFFT(PSI,NTRA,1)
      ENDIF
C     ==--------------------------------------------------------------==
C     == Apply the potential (V), which acts in real space.           ==
C     ==--------------------------------------------------------------==
      DO IB=1,NTRA
        DO L=1,NNR1
           PSIRE(L,IB)=RHOE(L)*PSIRE(L,IB)
        ENDDO
      ENDDO
C     ==--------------------------------------------------------------==
C     == Back transform to reciprocal space the product V.PSI         ==
C     ==--------------------------------------------------------------==
      CALL FWFFT(PSIRE,NTRA,1)
C     ==--------------------------------------------------------------==
C     == Decode the combination of wavefunctions (now multiplied by   ==
C     == the potential), back to those of states i and i+1. Multiply  ==
C     == also by the occupation numbers f(i) and other constants in   ==
C     == order to obtain the force on the electronic degrees of       ==
C     == freedom, stored in array FC.                                 ==
C     ==--------------------------------------------------------------==
      DO IB=1,NTRA
        I=IA+2*(IB-1)
        IS1 = I
        IS2 = I+1
        FI=F(IS1)*0.5
        IF(FI.EQ.0.0) FI=1.0
        IF(IS2.LE.NSTATE) FIP1=F(IS2)*0.5
        IF(FIP1.EQ.0.0) FIP1=1.0
        DO IG=1,NGW
          FP=PSI(NZH(IG),IB)+PSI(INDZ(IG),IB)
          FM=PSI(NZH(IG),IB)-PSI(INDZ(IG),IB)
          C2(IG,IS1)=-FI*
     *       ((TPIBA2*HG(IG))*C0(IG,IS1)+DCMPLX(DREAL(FP),DIMAG(FM)))
          IF(IS2.LE.NSTATE) C2(IG,IS2)=-FIP1*((TPIBA2*HG(IG))*
     *          C0(IG,IS2)+DCMPLX(DIMAG(FP),-DREAL(FM)))
        ENDDO
      ENDDO
      CALL TIHALT('      VPSI',ISUB)
C     ==--------------------------------------------------------------==
      RETURN
      END
C     ================================================================
```

FFT of wavefunctions in CPMD

We assume a simple cubic box with N^3 grid points. The wavefunction cutoff is 1/2 of the density cutoff. In reciprocal space only the values of grid points inside a sphere of radius N/4 are non-zero.
If $F(\omega) = 0$ then $f(t) = 0$.

Reciprocal space to direct space transform

full transform	sparse transform
N^2 1-D FFT	$\frac{\pi}{16}$ N^2 1-D FFT
N^2 1-D FFT	$\frac{1}{2}$ N^2 1-D FFT
N^2 1-D FFT	N^2 1-D FFT
3 N^2 1-D FFT	$\left(\frac{3}{2} + \frac{\pi}{16}\right)$ N^2 1-D FFT
1	0.57

Direct space to reciprocal space transform

full transform	sparse transform
N^2 1-D FFT	N^2 1-D FFT
N^2 1-D FFT	$\frac{1}{2}$ N^2 1-D FFT
N^2 1-D FFT	$\frac{\pi}{16}$ N^2 1-D FFT
3 N^2 1-D FFT	$\left(\frac{3}{2} + \frac{\pi}{16}\right)$ N^2 1-D FFT
1	0.57

The savings depend on the symmetry of the box.

Gradient corrections

$$\begin{aligned} E_{xc} &= \int F(n, \nabla n) d\Omega \\ V_{xc} &= \frac{\delta E_{xc}}{\delta n} = \frac{\partial F}{\partial n} - \sum_{\alpha=1}^{3} \frac{d}{dr_\alpha} \left[\frac{\partial F}{\partial(\nabla_\alpha n)} \right] \\ &= \frac{\partial F}{\partial n} - \sum_{\alpha=1}^{3} \frac{\partial}{\partial r_\alpha} \left[\frac{1}{|\nabla n|} \frac{\partial F}{\partial |\nabla n|} \frac{\partial n}{\partial r_\alpha} \right] \end{aligned}$$

The function F and its derivatives with respect to n and $|\nabla n|$ have to be calculated in real space. The derivatives with respect to r can be calculated most easily in reciprocal space.

- $$n(R) \stackrel{FFT}{\rightarrow} n(G)$$
- calculate $\frac{\partial n}{\partial r_\alpha} = iG_\alpha n(G)$
- $$iG_\alpha n(G) \stackrel{INVFFT}{\rightarrow} \frac{\partial n}{\partial r_\alpha}(R)$$
- calculate $|\nabla n(R)|$
- calculate F(n(R),∇n(R))
- calculate $\frac{\partial F(R)}{\partial n}$
- calculate $\frac{1}{|\nabla n(R)|} \frac{\partial F(R)}{\partial |\nabla n(R)|}$
- $$\begin{aligned} H^1(R) &= \frac{\partial F(R)}{\partial n} \stackrel{FFT}{\rightarrow} H^1(G) \\ H^2_\alpha(R) &= \frac{1}{|\nabla n(R)|} \frac{\partial F(R)}{\partial |\nabla n(R)|} \frac{\partial n}{\partial r_\alpha}(R) \stackrel{FFT}{\rightarrow} H^2_\alpha(G) \end{aligned}$$
- $$V_{xc}(G) = H^1(G) - \sum_{\alpha=1}^{3} iG_\alpha H^2_\alpha(G)$$
- $$V_{xc}(G) \stackrel{INVFFT}{\rightarrow} V_{xc}(R)$$

We can save some time by making use of the optimized real to complex transforms.

FFT of spherical functions

If we can write a function in real space as

$$f(\mathbf{r}) = f(r)Y_{lm}(\Theta, \Phi)$$

then its Fourier transform is

$$F(\mathbf{G}) = F(G)Y_{lm}(\bar{\Theta}, \bar{\Phi})$$

This we get by performing the integration

$$F(\mathbf{G}) = \int d\Omega f(\mathbf{r}) e^{i\mathbf{Gr}}$$

We make use of the identity

$$e^{i\mathbf{Gr}} = 4\pi \sum_{l=0}^{\infty} i^l j_l(Gr) \sum_{m=-l}^{m=l} Y_{lm}(\bar{\Theta}, \bar{\Phi}) Y_{lm}(\Theta, \Phi)$$

and we get

$$\begin{aligned} F(\mathbf{G}) &= \int d\Omega f(r) Y_{lm}(\Theta, \Phi) 4\pi \sum_{l=0}^{\infty} i^l j_l(Gr) \sum_{m=-l}^{m=l} Y_{lm}(\bar{\Theta}, \bar{\Phi}) Y_{lm}(\Theta, \Phi) \\ &= \int_0^{\infty} r^2 f(r) j_l(Gr) 4\pi i^l Y_{lm}(\bar{\Theta}, \bar{\Phi}) dr \\ F(G) &= 4\pi i^l \int_0^{\infty} r^2 f(r) j_l(Gr) dr \end{aligned}$$

where $j_l(z)$ are the spherical Bessel functions of the first kind.

$$\begin{aligned} j_n(z) &= \sqrt{\frac{\pi}{2z}} J_{n+1/2}(z) \\ j_0(z) &= \frac{\sin z}{z} \\ j_1(z) &= \frac{\sin z}{z^2} - \frac{\cos z}{z} \\ j_2(z) &= \left(\frac{3}{z^3} - \frac{1}{2}\right) \sin z - \frac{3}{z^2} \cos z \end{aligned}$$

Problem: For large G values $j_l(Gr)$ becomes a rapidly oscillating function and the integral is difficult to evaluate.
Solution: Transformation of variables.

- J.D. Talman, J. Comp. Phys. **29**, 35 (1978)
- J.D. Talman, Comp. Phys. Comm. **30**, 93 (1983)

Integrals of the form

$$g(k) = \int_0^\infty r^2 j_l(kr) f(r) dr$$

can be calculated by using the following transformation of variables

$$\begin{aligned} r &= e^\rho \\ k &= e^\kappa \\ \hat{g}(\kappa) &= g(e^\kappa) \\ \hat{f}(\rho) &= f(e^\rho) \end{aligned}$$

The integral in transformed variables is

$$\hat{g}(\kappa) = \int_{-\infty}^\infty j_l(e^{\kappa+\rho}) \hat{f}(\rho) e^{3\rho} d\rho$$

This can be rewritten as

$$\hat{g}(\kappa) = e^{(m-3/2)\kappa} \int_{-\infty}^\infty e^{(3/2-m)(\rho+\kappa)} j_l(e^{\kappa+\rho}) e^{(m+3/2)\rho} \hat{f}(\rho) d\rho$$

Here m is an integer that can be chosen arbitrarily $0 \leq m \leq l$. The new integral is a convolution and it is therefore possible to write

$$\begin{aligned} \hat{g}(\kappa) &= 2\pi e^{(m-3/2)\kappa} \int_{-\infty}^\infty e^{i\kappa t} M_{l,m}(t) \Phi_m(t) dt \\ &= 4\pi e^{(m-3/2)\kappa} \mathrm{Re} \int_0^\infty e^{i\kappa t} M_{l,m}(t) \Phi_m(t) dt \end{aligned}$$

where

$$\begin{aligned} \Phi_m(t) &= \frac{1}{2\pi} \int_{-\infty}^\infty e^{it\rho} e^{(m+3/2)\rho} \hat{f}(\rho) d\rho \\ M_{l,m}(t) &= \frac{1}{2\pi} \int_{-\infty}^\infty e^{-it\rho} e^{(3/2-m)\rho} j_l(e^\rho) d\rho \end{aligned}$$

The basic integral can therefore be evaluated by carrying out the Fourier transform to calculate $\Phi_m(t)$, multiplying by $M_{l,m}(t)$ and carrying out the

Fourier transform for $\hat{g}(\kappa)$. All the integrals can be evaluated by fast Fourier transforms if we use equidistal grids with the same spacing for κ and ρ and the corresponding grid for t. Equidistal grids for the transformed variables correspond to logarithmic grids for the original variables r and k.
The integral for $M_{l,m}(t)$ can be evaluated analytically as

$$\begin{aligned} M_{l,m}(t) &= \frac{1}{2\pi}\int_0^\infty r^{1/2-m-it} j_l(r)dr \\ &= (8\pi)^(- 1/2)\prod_{j=1}^{p}(j-1/2-it) \\ &\quad \prod_{j=1}^{l}(2j-l+m-1/2+it)^{-1} \\ &\quad \left[\cos\frac{p\pi}{2}e^{i(\Phi_1-\Phi_2)} + \sin\frac{p\pi}{2}e^{i(\Phi_1+\Phi_2)}\right] \end{aligned}$$

where p = l - m and

$$\begin{aligned} \Phi_1 &= \arg\left[\Gamma(1/2-it)\right] \\ \Phi_2 &= \tan^{-1}(\tanh(\pi t/2)) \end{aligned}$$

It was observed that for the choice m = 0 results accurate at large k values are obtained, whereas if m = l, the results are accurate for k close to zero. To obtain accurate values for all k one carries out the calculations for m = 0 and m = l and joins the data at an optimal point. For l = 0 an integration of the initial equation leads also to accurate results for small k.