

How to use POTGEN, SQUARER and FKPMC codes

D. M. Ceperley

October 16, 2001

Contents

1	Introduction	1
2	Input	2
2.1	Input formatting	2
2.2	Input Commands for POTGEN and SQUARER	2
3	POTGEN output	5
4	Sample input and output for POTGEN	5
4.1	Helium	6
5	SQUARER Output	6
6	Feynman-Kacs (MC) evaluation of the density matrix	7
7	Graphical Ouput with SMongo	9
8	Tests	9

1 Introduction

One starts the study of a new physical system with the specification of the potential energy function done by the program POTGEN. If the potentials are very non-linear it is important to use the best action to make the PI calculations feasible. We have found a very convenient choice to be the exact action for a pair of particles. This is what the program SQUARER calculates (as well as some related quantities.) Another code FKPMC does a Monte Carlo evaluation of the action for 2 or more atoms in order to test SQUARER or to see how good the pair action is for more than 2 atoms.

SQUARER computes the two-atom density matrix, or equivalently, the action of one particle in an external potential. The density matrix is defined as the solution of the Bloch equation:

$$-\frac{\partial \rho(\mathbf{r}, \mathbf{r}'; \mathbf{t})}{\partial t} = [-\lambda \nabla^2 + v(\mathbf{r})] \rho(\mathbf{r}, \mathbf{r}'; \mathbf{t}) \quad (1)$$

for a potential $v(\mathbf{r})$. Here \mathbf{r} can be either a 1-, 2- or 3-dimensional vector. The initial condition is $\rho(\mathbf{r}, \mathbf{r}'; \mathbf{0}) = \delta(\mathbf{r} - \mathbf{r}')$. The action is defined as:

$$u(\mathbf{r}, \mathbf{r}'; \mathbf{t}) = -\ln[\rho(\mathbf{r}, \mathbf{r}'; \mathbf{t})/\rho_0(\mathbf{r}, \mathbf{r}'; \mathbf{t})] \quad (2)$$

where ρ_0 is the free particle density matrix ($v(\mathbf{r})=0$). In the *primitive approximation* (good at high temperature ($T=1/t$)) the action is:

$$u(\mathbf{r}, \mathbf{r}'; \mathbf{t}) = \frac{\mathbf{t}}{2} [\mathbf{v}(\mathbf{r}) + \mathbf{v}(\mathbf{r}')]. \quad (3)$$

The name SQUARER comes from the algorithm to compute the exact action. It works by constructing a high temperature “semi-classical” approximation and then numerically squaring

this approximation to get down to low temperature. You are advised to read pages 318-320 of Ceperley (1995) to get a general mathematical overview of density matrices and how the various methods work. What is emphasized here are how to use the program and comprehend the output.

SQUARER calculates the action $u(\mathbf{r}, \mathbf{r}'; \mathbf{t})$ and its derivative with respect to t ($u_t(\mathbf{r}, \mathbf{r}'; \mathbf{t})$) used to estimate the internal energy respectively. Internally SQUARER uses a partial wave decomposition to represent the angular variables, but for output it converts to a representation in distances only. These results are written to files *name.dmu* and *name.dme* respectively. These files need to be recomputed when you change the potential, mass of the atoms, or the time step “ t ”.

The set-up of the potential is done in the separate program POTGEN. One then uses the output of POTGEN to construct the density matrix with the SQUARER code. Each pair interaction is treated separately. For n different chemical species, you will need $n(n+1)/2$ different interaction potential and files, unless some are related by symmetry. I will first describe the input of both programs together.

2 Input

2.1 Input formatting

The input to the programs is keyword driven. A new command always starts a line with a *keyword*, followed by parameters. Some commands are relevant to only a specific program, others to several different programs. Keywords not understood or irrelevant to a given program will be ignored. The general syntax is:

1. A command begins with a keyword. Keywords are always UPPER case. Case is sensitive. Only one keyword per line. No continuations are allowed.
2. Parameters must be in the correct order. They can be separated by spaces or commas.
3. Some parameters at the end of a command line might be optional. In that case I put a left bracket ([). For example: WALLS x0 [x1 means that the parameter x1 is optional. If it is not present a default value will be supplied.
4. Text after a colon or semicolon is ignored. You can put comments in the input this way
5. Real numbers must have a '.' (decimal point) on some computers.
6. Character variables cannot contain blanks, commas, colons or semicolons.

2.2 Input Commands for POTGEN and SQUARER

The input (and output) file to POTGEN and SQUARER can contain the following essential or optional directives:

1. GRID (essential for POTGEN and SQUARER)

The grid for the potential is specified as input to POTGEN and is passed later to SQUARER, OCSET and UPI. It cannot be later changed except by running the entire sequence of codes again.

A grid is a monotonically increasing, smooth function, mapping the radial coordinate r from $r_1 \leq r \leq r_n$ to x where $1 \leq x \leq n$, $n > 1$ and $r_1 < r_n$. The mapping $f(r) = x$ has to be something that is fast to compute since it is in the heart of the PIMC code. The linear grid $x = x_0 + ar$ is fastest (since we already have determined r .) Because of the interpolation schemes used the grid must be analytic without singularities in important places. The inverse mapping $r = f^{-1}(x)$ is occasionally needed although it need not be fast. The reasons for having a non-linear grid are to handle singularities, such as the Coulomb potential has at the origin, to put more grid points where the potential is rapidly changing so that SQUARER needs fewer points (both its execution time and memory scale as n^3) and to cut down on memory in PIMC (which goes as n).

$n = 200$ is typical, especially on starting a project. If a distance is greater than r_n or less than r_1 is needed, normally the code will *extrapolate*. There is an option provided to change this behavior. WALLS ensures that the density matrix exactly vanishes outside the walls.

The GRID command is:

```
GRID n type r1 rn
```

The types of grids that are currently programmed are:

- linear: $r_i = r_1 + (r_n - r_1)(i - 1)/(n - 1)$.

```
GRID n LINEAR r1 rn
```

- logarithmic: $r_i = r_1 * (r_n/r_1)^{(i-1)/(n-1)}$.

```
GRID n LOG r1 rn
```

- Inverse grid: $r_i = cnst + cnst/(i - cnst)$

```
GRID n INVERSE r1 rn y
```

The parameter $y > 0$ is the relative expansion of the grid at r_1 . That is $y = 0.1$ means there is a 10 times finer spacing around r_1 and a 10 times coarser spacing around r_n than one would have for a linear grid. $y = 10$ would be the opposite. $y = 1$ corresponds to a linear grid and cannot be used.

- power grid : $r_i = a + b(i - x_0)^d$

```
GRID n POWER r1 rn x0 d
```

restrictions: $x_0 < 1, d \neq 0$ This is a very good grid for Lennard Jones like potentials using $d = -0.5$. By placing x_0 slightly less than unity, one can put in many grid points at small r . $d = -1$ corresponds to the INVERSE grid. The disadvantage is that an exponentiation is made so it is slow.

A word on implementation: setgrid.f computes the grid, interp.f does the interpolation.

2. POT (essential for POTGEN, not used in SQUARER)

This command is used by POTGEN to identify a preprogrammed potential and to add a table at the end of the *name.dm* file. It is anticipated that you will have to change some things in POTGEN since we can't program all possible potentials. New potentials can be programmed in the subroutine *ipot*. The first POT argument is distance where the potential is truncated and shifted up and the second the name of the potential.

The following potentials are pre-programmed in POTGEN:

- helium-helium Aziz potential

```
POT rc HEDF2
```

- helium-neon interaction

```
POT rc HeNe
```

- hydrogen-hydrogen Silvera-Goldman potential

```
POT rc H2H2
```

- LJ: Lennard-Jones 6-12.

$$v(r) = 4\epsilon[(\sigma/r)^{12} - (\sigma/r)^6] \quad (4)$$

```
POT rc LJ eps sigma
```

- POWER: Any pure inverse power.

$$v(r) = 4\epsilon(\sigma/r)^\alpha \quad (5)$$

POT rc POWER epsilon sigma alpha

(f) WALL A helium wall with a possible centrifugal term.

$$v(r) = v(rwall - r) + cent/r * 2 \quad (6)$$

POT rc WALL rwall [cent

3. UNITS (optional for all programs)

This is passed through all following programs and not used (future versions of POTGEN may automatically convert units.) It is provided to remind the user that the energy and length units must be kept consistent. Whatever units are chosen for energy and length will be used for the temperature units, size of the simulation box, number density, etc. For example, it is assumed that λ (on the TYPE) line is in units of energy*length². For multi-component systems, the units for each pair interaction must be the same.

UNITS energyunit lengthunit

4. TYPE (essential for SQUARER, not used in POTGEN)

This names the particles that are interacting and for which you want to calculate the pair density matrix and gives their properties. The λ (the inverse mass) is needed by SQUARER. One should have *two* TYPE commands for SQUARER since SQUARER deals with pair interactions. If only one is present, then SQUARER assumes that one is dealing with interactions between equally massive particles and it puts in an extra one for you.

Input format:

TYPE name lambda

name is the name of the particle. In UPI it will be used to label files (the input and output files to UPI, averages pair correlations, etc.) $\lambda = \hbar^2/2m \geq 0$ is the quantumness parameter (in units of energy*length²). If $\lambda = 0$ one assumes that particle is classical particle. In SQUARER the reduced mass parameter; $\lambda_e = \lambda_1 + \lambda_2$ is used. If $\lambda_e = 0$, a classical density matrix is computed by SQUARER: $u(r) = \tau * v(r)$. Useful conversion: $\lambda = 24.254/m$ where m is in atomic mass units and λ is in units of K Å².

5. SQUARER (essential for SQUARER, not used by other programs).

This must be the last input to POTGEN.

SQUARER tlow ndim norder nl nsquare

All parameters are needed. They are defined as:

- *tlow* is the lowest temperature at which the density matrix is generated.
- *ndim* is the spatial dimensionality in the range (1,2,3). *ndim* = 1 is for the interaction of a particle with a plane. *ndim* = 2 for 2 particles in 2D or for a 3D particle interacting with a line or cylinder. *ndim* = 3 is for 2 particles in 3D. *ndim* = 1 will cause *nl* = 0 and *norder* = 1 to be set.
- *norder* is the highest order of the polynomial fit. *norder* = 0 means to use diagonal only. *norder* = 1 will include terms *s* and *z* etc. *norder* = 3 is typical for good accuracy.
- *nl* is the number of partial waves. A typical value is 30.
- *nsquare* are the total number of squarings to reach lowest temperature. Hence the squaring process begins at a temperature of $tlow 2^{nsquare}$. *nsquare* = 0 is a flag for the primitive approximation. *nsquare* ≥ 0 .

6. DUMP (optional for SQUARER)

DUMP [11 , 12 ...

If this keyword is present, output of the partial wave s (11 i, 12 ...) will be written to files *fort.11* etc. These files are for making graphs.

7. WALLS. (Optional for SQUARER, not needed in POTGEN.)

This command zeros the density matrix for $x_0 \leq r$ and for $r \geq x_1$. This can be useful to prevent exponential overflow. For $ndim > 1$, SQUARER always sets $x_0 \geq 0$. The default for $x_1 = +\infty$. The wall locations must be outside the grid.

WALLS [x0 [x1

8. ONLY. (Optional for SQUARER, not used in POTGEN.)

This command forces the fitting to be a function of s^2 only. The variable z^2 is not used. This is the default for $ndim = 1$ (since $s^2 = z^2$) but overrides the default for $ndim > 1$. This command is useful for coulomb interactions since one knows from symmetry that there is no z^2 dependence. PUPI will run slightly quicker if it does not have to compute the z^2 terms but for general potentials, accurate actions will not be obtained.

ONLY

3 POTGEN output

The output of POTGEN consists of the file *name.dm* containing:

- A copy of all of the input to POTGEN. This way one can pass along input to SQUARER, the setup program and UPI.
- The POTTAIL line. This contains an asymptotic expansion of the potential in inverse powers of r . This is used in the setup programs to compute “tail” corrections (the contribution of the potential coming from outside the box). The data on this line are a) The shift in the potential at cutr b) the number of terms in the expansion that follow c) pairs of exponents and coefficients. If absent no tail correction will be made.

POTTAIL n alpha_1 C_1 ... alpha_n C_n

- The CITATION command. The gives a literature citation if it is available. This is only used to document the density matrix file.

The potential table is in the file *name.pot*. The three columns contain 1) r_i , 2) $v(r_i)$ and 3) maximum interpolation error at that grid point. The last is to help you intelligently choose a grid spacing. One should have an error orders of magnitude smaller than the temperature of the simulation. Say $error(r) < 0.01T$ for r in the physical range of interest.

You do not have to use the POTGEN program to make the input to squarer. All SQUARER cares about is if the data is in the right place. (and only the second column is read by SQUARER.)

4 Sample input and output for POTGEN

There are some sample inputs and outputs to squarer in the directory *test*.

4.1 Helium

Here is the input (He4.60.dm) for a pair of ^4He atoms. Units are Kelvin and Angstroms. The grid is linear from 0.2 Å to 8.0 Å and contains 60 grid points. We use the HEDF2 (Aziz potential). It is zeroed and shifted up at 6.0 Å. (The grid is really too small for accurate calculations but useful for testing.)

SQUARER will generate density matrices at 4 temperatures: 10K, 20K 40K and 80K. For each of them it generates diagonal, linear and quadratic off-diagonal approximations. The atoms are in 3 dimensions. 50 partial waves are used. 14 squarings are done to reach the lowest temperature of 10K.

```
UNITS K A
TYPE He4 6.059615
TYPE He4 6.059615
GRID 60 LINEAR .2 8.
POT HEDF2 6.
SQUARER 10. 4 3 3 50 14
```

The resulting file *He4.dm* is:

```
UNITS K A
TYPE He4 6.059615
TYPE He4 6.059615
GRID 60 LINEAR .2 8.
POT HEDF2 6.
SQUARER 10. 3 2 30 14
CITATION:R. A. Aziz, A. R. Janzen, M. R. Moldover, Phys. Rev. Letts. 74, 1586 (1995)
POTTAIL -0.23509E+00 3 -6.00 -0.10131E+05 -8.00 -0.27398E+05 -10.00 -0.99775E+05
```

The POTTAIL line shows the potential was shifted up by 0.235 K inside 6 Å and has an expansion: $c_6 r^{-6} + c_8 r^{-8} + c_{10} r^{-10}$ at large r . The potential is on file He4.60.pot. The interpolation error at the zero value of the potential is about 0.06K.

5 SQUARER Output

First let me list the different files output by squarer.

1. The output file *name.dmo* gives some interesting output that I detail next.
2. The output file *name.dmu* contains the action. The first column is the radial grid, the second the rms error in converting to the polynomial representation, the next column is u00, then u01, u11 , u02 , u12, u22 etca (depending on the order).
3. The output file *name.dme* contains the derivative of the action u_t in exactly the format of the .dmo file.

The input parameters *nl*, *n* and *norder* should be carefully chosen. If they are too small incorrect results will be obtained; if they are too large the execution time of SQUARER or UPI will be excessive. You can look at the output files to get some ideas for how things went.

If SQUARER stops it is likely that one of the parameters has been set too large for the corresponding array size and a parameter in SQUARER must be increased and SQUARER remade. These parameters are called *ml*, *m_x* and *morder*, respectively. The current values and the dimensioned value are shown on the standard output.

The following are some general ideas of how to pick the adjustable parameters.

1. The grid size *n*. All you can do here is
 - try different grids and see if the action and derivatives are fairly stable. Look at the error of potential energy in fname.pot.

- Look at $u(1)$ and $u(n)$. $u(1)$ should be smaller than about $1.e-5$ if you have a hard core like potential. This determines $r1$. $u(n)$ should be smaller than 0.001 . Otherwise $cutr$ is too small.

The following example shows at the edges of the grid u and u_t are quite small on the diagonal:

```
r    .20000E+00 rho    .13501-268    .34433-263
r    .75000E+01 rho    .10000E+01    -.12791E-05
```

2. Number of partial waves: nl . To adjust this parameter look at the *name.dmo* file for last line of the form:

```
partial waves needed = 31
```

This means that for an accuracy of $1.e-4$ only 31 partial waves were needed. (But note that this number must be less than nl). This information can save you time in the future SQUARER runs as you can use this value for future runs if you had chosen it too large. But if equals nl you should increase it in the future runs.

3. Order of approximation, $norder$. To adjust this parameter you look at the *name.dmu* file and *name.dmu* file. The second column has the rms error as a function of the the distance, (What is computed is the error in the pair action caused by going from the partial wave expansion to the polynomical expansion. Of course there are errors coming from 3 and higher-body interactions).

It is best if the maximum error on the action (file *name.dmu*) is less than 0.01 in regions of physical interest. The file *name.dmo* has the summary errors for each

```
order =    1 chimax    .41379E-01    .20154E+02
order =    2 chimax    .16580E-02    .52513E+00
order =    3 chimax    .20204E-04    .40512E-01
```

The first column is the maximum error of u (times the exponentiated action) , the next in u_β . So with the diagonal approximation ($order = 1$) u has a maximum error of 0.041 . I would consider this a pretty large error. Errors in the energy in the end point approximation may be as much as $20.K$. With the best approximation ($norder = 2$) the errors are much smaller, $2E-5$ in the action and $0.04 K$ in the energy. However UPI will be slower for $order = 2$.

6 Feynman-Kacs (MC) evaluation of the density matrix

A good way to test the output of squarer is to use the program FKPMC. This reads in the *name.pot* output of POTGEN and computes the action at specific points. It is not so convenient to use this to generate tables since it is slow and the results are noisy but it doesn't make a breakup either into partial waves or into a grid so it does produce a good global check. An important use of FKPMC is to calculate density matrices for more than two particles to see how good the pair density matrix is for a many-body system. The only systematic errors in FKPMC are that: 1) It calculates the potential from a cubic spline of the values on *name.pot*. 2) It calculates the action using paths of $2, 4 \dots 2 * ndiv$ steps and extrapolates to $\tau = 0$ assuming a τ^2 dependence. If τ is too large this could give an error.

Input to FKPMC is in the file *fkpmc.in*. It is format free and does not follow the normal keyword driven format. Here is an input for two helium atoms:

He4_60	Name of the potential file He4_60.pot
TYPE He4 6.059615	TYPE card only the third parameter lambda is used
GRID 60 LINEAR .2 8.	GRID for the potential used in the He4_60.dm file
10.	Temperature to compute the action.
2 500 4 64	Number of particles, MC steps, levels and blocks.
0. 0. 0.	r1 initial.
0. 0. 0.	r1 final.

```

3.1085 0. 0.          r2 initial.
3.1085 0. 0.          r2 final.

```

The computer time scales as $2^{n_{\text{levels}}} n_{\text{steps}} n_{\text{blocks}}$. $n_{\text{blocks}} = 64$ is a good choice as it provides enough blocks to get good error estimates and it vectorizes well.

Here is the resulting output:

```

Begin Feynman-Kacs density matrix computation
Author: D. Ceperley, University of Illinois. Version  of August 7,1997
1 CMD:TYPE He4 6.059615
1 CMD:GRID 60 LINEAR .2 8.
GRID          60 0.200000000000000000      8.0000000000000000
temperature=  10.0000000000000000
number particles=          2
spatial dimension=        3
particle   1 initial position    0.00000    0.00000    0.00000
particle   1 final position      0.00000    0.00000    0.00000
particle   2 initial position    3.10850    0.00000    0.00000
particle   2 final position      3.10850    0.00000    0.00000
Potential taken from file He4_60
finished reading potential
random number seed set to 870991468      0
diffusion length=  1.906769257146758    |R1-R2|=  0.0000000000000000E+00
V(R1)= -10.03536588076732    V(R2)= -10.03536588076732
number of blocks=          64
number of levels=          4

beginning Monte Carlo, nsteps*nblocks=    32000
level Action      error
  0 -0.1004E+01    0.0000E+00
  1 -0.6272E+00    0.2569E-02
  2 -0.3759E+00    0.3412E-02
  3 -0.2437E+00    0.3649E-02
  4 -0.1923E+00    0.3833E-02
extrp -0.1752E+00    0.3991E-02
bias              -0.7228E-05
rel.ex            0.4293E+01

```

The line labeled “extrp” contains the resulting extrapolated estimate of the action and (-0.175 ± 0.004) . The line “bias” shows errors coming from taking the logarithm and other non-linear effects. The line “rel.ex” shows the extrapolation relative to the error bar. If this is much greater than one, you should increase the number of levels (unless you are confident that you are in the asymptotic τ^2 regime).

Here is the input which produced figure 35 and 47 in Ceperley (1995), for 3 helium atoms in an equilateral triangle.

```

He4_60
TYPE He4 6.059615
GRID 60 LINEAR .2 8.
40.
3 2000 5 64
0 0 0
0 0 0
2.88 0 0
2.88 0 0
1.44 2.494 0
1.44 2.494 0

```

The resulting output is:

```

Begin Feynman-Kacs density matrix computation
Author: D. Ceperley, University of Illinois. Version of August 7,1997
1 CMD:TYPE He4 6.059615
1 CMD:GRID 60 LINEAR .2 8.
GRID          60 0.2000000000000000      8.000000000000000
temperature=  40.00000000000000
number particles=      3
spatial dimension=     3
particle  1 initial position    0.00000    0.00000    0.00000
particle  1  final position     0.00000    0.00000    0.00000
particle  2 initial position    2.88000    0.00000    0.00000
particle  2  final position     2.88000    0.00000    0.00000
particle  3 initial position    1.44000    2.49400    0.00000
particle  3  final position     1.44000    2.49400    0.00000
Potential taken from file He4_60
finished reading potential
random number seed set to 870992119      0
diffusion length= 0.9533846285733790    |R1-R2|= 0.000000000000000E+00
V(R1)= -30.80330896378227    V(R2)= -30.80330896378227
number of blocks=      64
number of levels=      5

beginning Monte Carlo, nsteps*nblocks=    128000
level Action      error
  0 -0.7701E+00    0.0000E+00
  1 -0.3891E+00    0.1122E-02
  2 -0.2260E+00    0.1352E-02
  3 -0.1744E+00    0.1497E-02
  4 -0.1608E+00    0.1531E-02
  5 -0.1571E+00    0.1530E-02
extrp -0.1558E+00    0.1532E-02
bias          -0.1151E-05
rel.ex        0.8191E+00

```

7 Graphical Ouput with SMongo

There is a macro and script to plot the density matrix file and errors with smongo. The procedure to plot them is to start up Supermongo and type:

```

macro read plotdm
plotk [u, e] name r1 r2
pot name r1 r2
error [ u , e] name r1 r2

```

Here the plotting range is from r1 to r2. *k* is refers to *norder* and must be 1 or 2. *pot* plots the potential. *error* plots the chisquare (for u or e).

Associated files with this program are: smplot, plotdm

8 Tests

Hopefully SQUARER will always work correctly for you. If you suspect it is malfunctioning, it is good to have some ideas of where to start testing. Some ideas of tests are:

1. Use the DUMP option to look at the diagonal action and the partial waves to see if they are smooth. One can look at the various partial wave components to see if they are at all similar to what comes from on eigenfunction expansion. At low temperature we should get: $u(r,l) \approx \beta E - \log(\psi_l)$ We will have to develop a eigenvalue solver to use this information.

2. Check the derivatives by hand. If you compute the density matrix at 2 nearby temperatures you can compute u_t by hand. This is useful to check
3. See how things depend on the grid, number of partial waves and squarings. Things can get unstable if they are chosen too big.
4. You can turn off the potential (no POT command). The free particle action should be close to zero. Typical accuracies are 1.e-10.

References

- [1] Ceperley, D. M., 1995, Rev. Mod. Phys., **67**, 279.
- [2] Klemm, A. D., and R. G. Storer, 1973, Aust. J. Phys. **26**, 43.
- [3] Pollock, E. L., 1988, Comput. Phys. Commun. **52**, 49.
- [4] Pollock, E. L., and D. M. Ceperley, 1984, Phys. Rev. B **30**, 2555.
- [5] Storer, R. G., 1968, J. Math. Phys. **9**, 964.