ASA layer Green's function package (v6.14)

Mark van Schilfgaarde

28 Oct, 2004

1 Introduction

This package implements the ASA local spin-density approximation using Green's functions (GF) in a special layer geometry. It adds a program 'lmpg' to the ASA suite, which plays approximately the same role as the LMTO-ASA band program 'lm' or the crystal Green's function program 'lmgf'.

lmpg is similar in most respects to the crystal GF package, except for some significant complications that arise from treatment in a layer geometry. In **lmpg**, there is a 'special direction', which defines the layer geometry, and for which the GF are generated in real space. It is specified by the third of the primitive lattice translation vectors (token PLAT). In the other two directions, Bloch sums are taken in the usual way; thus for each qp in the parallel directions, the hamiltonian becomes one-dimensional and is thus amenable to solution in order-N time in the number of layers.

In the layer geometry, the material consists of an active, or embedded region, which is cladded on the left and right by left and right semi-infinite leads. The crystal (or bicrystal) is partitioned into slices, or principal layers (PL), in the 'special direction'. This is done because the hamiltonian H is short-ranged: if each PL is thick enough so that H only connects neighboring PL, the computational effort scales linearly with the number of PL. Actually **lmpg** just assumes each PL is thick enough so that H is tridiagonal in the PL representation. The principal layers are defined by you in (see token RMAX=, Sec. 2). It is your responsibility to see that each PL is thick enough so that H connect to only nearest-neighbor PL on either side.

lmpg has to deal with some of the complications arising from how the endpoints are treated. The first PL is treated as the 'left endpoint' and the last PL are treated as the 'right endpoint.' The first PL and last PL are replicated to the left and right, respectively, and extend outside of the embedding region. (Both are replicated implicitly an infinite number of times to form the semi-infinite leads). In the 'left' case a replica each site belonging to first PL is shifted 'to the left' by PLATL; similarly, a replica of each site belonging to last PL are are shifted 'to the right' by PLATR, so that we have

$$\ldots \mid \overbrace{\mathrm{L}-\mathrm{replica}}^{\mathrm{PLATL}} \mid \overbrace{\mathrm{PL}\ 0 \mid \mathrm{PL}\ 1 \mid \mathrm{PL}\ 2 \mid \ldots \mid \mathrm{PL}\ n{-}1}^{\mathrm{PLAT}} \mid \overbrace{\mathrm{R}-\mathrm{replica}}^{\mathrm{PLATR}} \mid \ldots$$

See Sec. 2 for how **lmpg** reads PLAT, PLATL, PLATR.

It should be clear that the embedding region should be constructed so that the PL adjacent to the leads are already bulk-like. The C- region should be made large enough so that charge densities of the PL near the edges of the C-region (e.g. PL 0 and 1, and PL n-1 and n) rather closely resembles densities of the semi-infinite (bulk) leads. This point is discussed later.

As we discuss later, the GF of the embedding region is computed from the (surface) GF of the L-replica and the (surface) GF of the R-replica. Thus space is partioned into three regions, the left (L-) end region, the right (R-) end region, and the center, embedding (C-) region. The end regions are treated specially in two contexts:

1. Surface Green's functions for the end regions are needed to supply the boundary conditions for the embedding Green's function

2. **lmpg** requires special treatment for the electrostatics joining the three regions.

1.1 Green's functions for the end regions

By its construction the PL of each end (L- and R-) region would, if repeated throughout all space, constitute a periodic solid. **lmpg** has a special branch (specified by token MODE=2, Sec. 2) for the L- and R- PL that enable it to generate the self-consistent left- and right- GF for each corresponding periodic solid. This is needed to make the potential of each end region. Note that it is assumed to be bulk-like. The GF should be the same as that generated by 3D Green's function program **lmgf**, except that this GF has a mixed-real and k-space representation, and here separate Green's functions and potentials are needed for each end region. Also, owing to the mixed mixed-real and k-space representation, the methodology for constructing G is different. We will call the periodic solid of repeating L- PL the "bulk" crystal of the L- region; similarly for the R- PL. Thus, there is a well-defined "bulk" GF and potential the L- PL, and one also for the R- PL.

Once the L- and R- bulk potentials are made, the surface Green's function can be computed, which is needed to supply the proper boundary conditions to generate the GF in the embedding region.

1.2 Self-consistency and charge neutrality

A Green's function (or any band) method requires integration to the Fermi level, which is determined by charge neutrality. Also in general, the electrostatic potential is determined up to an arbitrary constant; but specifying one determines the other. In the band code, we supply (or assume) the constnat, and find the corresponding Fermi level. In the GF programs we indirectly fix this constant potential by specifying the Fermi level as an input. Requirement of charge neutrality fixes the constant potential shift. Note that the GF have a complication not present in a band program, where the entire spectrum is computed: the constant shift must be determined by an iterative procedure. In the crystal GF case, it is iteratively determined by a Pade approximant (see lmgf documentation). Metals and nonmetals are distinguished in that in the latter case, there is no DOS in the gap and therefore the Fermi level (or potential shift, if the Fermi level is specified) cannot be specified precisely. As we will describe, something similar happens in the layer case, but it is a little more complicated. The Fermi level and constant potential are stored in array **vshft**, and permanently on disk in file **vshft** described below and in the documentation found in the source code **iovshf.f**. Actually **lmpg**'s freedom to shift potentials is more general and can accomodate potential shifts at separate sites, useful in non-self-consistent or limited self-consistent calculations.

The layer GF case is complicated by the partioning into three distinct regions. Self-consistency, and therefore determination of potential proceeds differently depending on whether one is computing the potential for the bulk L- and R- (MODE=1) or for the layer system $(\ldots\,|\,\mathrm{L}\,|\,\mathrm{C}\,|\,\mathrm{R}\,|\ldots)$ (MODE=1). Computing the bulk potential for the L- and R- regions (MODE=2), is quite analogous to the crystal GF, albeit for two independent regions L and R. Periodic boundary conditions are assumed separately; each case the end layer is assumed to be a periodic solid and the (periodic) Green's function is computed for it. Thus the potential in each end layer may be shifted independently by a constant shift V_L or V_R. Self-consistency proceeds analogously to the crystal GF, independently for the regions L and R. Constant potential shifts are determined independently for each layer in the same way as the in crystal GF code; the potential shift is computed that satisfies charge neutrality for the corresponding periodic solids; see lmgf documentation, for further details. If the L- (R-) PL is a metal, the constant potential shift is not adjustable, because the Fermi level is specified at the outset. Note, however, that the Fermi level is only sharply defined for a metal; thus it is important to distinguish metal and nonmetal cases independently in the two layers. You can set them with tokens LMET= and RMET= in the PGF category; see Sec. 2. These tokens play the role of the METAL= token for the crystal GF code.

Charge neutrality in the layer case $(\ldots\,|\,\mathrm{L}\,|\,\mathrm{C}\,|\,\mathrm{R}\,|\ldots)$ is more complicated. It should be satisfied independently in the each of the L-, C- and R- regions. In practice, we *assume* that the density in the L- and R- end regions is bulk-like and does not change once it has been calculated (MODE=2). Thus changes in the density are confined to the C- region. The C- region has to be charge-neutral because the L- and R- are already neutral, and the entire $(\ldots\,|\,\mathrm{L}\,|\,\mathrm{C}\,|\,\mathrm{R}\,|\ldots)$ system must be neutral. The program proceeds by finding a shift that satisfies charge-neutrality in the C- region and doesn't worry about the rest. This is reasonable since we assume at the outset that the C- region has enough PL near L- and R- large enough to allow the density to be bulk-like, no charge should spill into the L- and R- regions by construction. Thus, when computing the Green's function of the $(\ldots\,|\,\mathrm{L}\,|\,\mathrm{C}\,|\,\mathrm{R}\,|\ldots)$ system in practice, the layer code selects the shift V_C so as to eliminate the deviation from charge neutrality in the C- region, following the method of the crystal code **lmgf**. Only sites in the C- region are shifted by V_C; the potentials and charges in the L- and R regions are left untouched. Once V_C is found and the corresponding Green's function is generated, **lmpg** returns to the sphere program where it computes the potential functions for the updated sphere charges and moments, computes the electrostatic potential (see below) for the $(\ldots|\mathrm{L}|\mathrm{C}|\mathrm{R}|\ldots)$ system, and begins another pass in the self-consistency cycle.

Because of deviations from self-consistency, and also because of finite-size effects discussed above, there can be some deviation from charge neutrality in the end regions. **lmpg** will generate the GF in each end PL, so that the deviation from neutrality may be computed. (You must have the 'bigemb' option set for **lmpg** to generate this information.) The sphere charges are shown in the table headed by the lines:

```
PGFASA: integrated quantities from G
    PL      D(Ef)      N(Ef)       E_band      2nd mom      Q-Z
```

and deviations of the end regions from charge neutrality are summarized at the end of this table in lines similar to the following:

```
Deviations from charge neutrality:
  Left end layer       0.003965
  Right end layer      0.003965
```

However, **lmpg** does not use this information; instead it keeps the densities as computed for the bulk L- and R- crystals. If these charges are not small, your active region should be enlarged.

1.3 Electrostatics

At self-consistency there is a unique potential defined by the electrostatic potential from the charges at each site and a global constant potential shift V_C that shifts the entire system. This shift makes makes each region charge neutral and possesses within the C- region a dipole that will exactly align the Fermi levels in the L- and R- regions, as we now describe.

For now let's restrict ourselves to the case when both L- and R- regions are metals. We can form compute electrostatic potentials in the L- and R- regions by two different constructions:

1. Electrostatic potentials in L- and R- may be computed as in MODE=2, that is by assuming L- and R- are bulk solids with periodic boundary conditions in the respective L- and R- regions. In each case the potential is completely fixed by charge neutrality.

2. Electrostatic potentials in L- and R- may also be computed from solution of the potential of the entire $(\ldots|\mathrm{L}|\mathrm{C}|\mathrm{R}|\ldots)$ system. This potential is adjustable up to some constant shift.

In practice these two constructions produce different potential in the L- and R- regions. (Indeed, one might choose the constant shift that "best reconciles" the mismatch in these two constructions. Versions of lmpg earlier than 6.14 did something like this. The present version choose the constant shift so as to satisfy charge neutrality in the C- region, as was discussed in the previous section.)

These potentials might be mismatched for two distinct reasons. One error can arise because the density is not self-consistent. More exactly the density doesn't possess the requisite dipole, so that a global constant shift that "best reconciles" the potential mismatch as constructed by the two methods above is different from the one that "best reconciles" the mismatch on the right. Finite-size effects (a C- region with insufficient PL near the end regions), is another source of error.

Recall that **lmpg** freezes the potentials in the L- and R- regions. Thus, only the central region is affected by the shift V_C because the end PL shouldn't be shifted at all. **lmpg** does print out the electostatic potentials computed by the two different constructions, and summarizes the deviations in a table similar to the following:

```
Deviations in end potentials:
region met  <ves>Bulk   <ves>layer     Diff      RMS diff
 L      T    0.055210    0.058192   0.002982    0.011265
 R      T    0.055210    0.058192   0.002982    0.011265

RMS pot difference in L PL = 0.000127  in R PL = 0.000127  total = 0.000127
vconst that minimizes potential mismatch to end layers =  0.058835
vconst is now (estimate to satisfy charge neutrality)  =  0.061817
difference                                             = -0.002982
```

The top table compares the average electostatic potential in and end region computed from the bulk geometry, and computed in the $(\ldots\,|\,\mathrm{L}\,|\,\mathrm{C}\,|\,\mathrm{R}\,|\,\ldots)$ geometry. The later numbers compare the potential that "best reconciles" the the methods of computation to the potential shift the program will actually use. Note that the potential shift you actually should use is the one that meets charge-neutrality in the C- region; indeed **lmpg** will find this shift on its own if you invoke it in self-consistent mode (section 2).

When getting started, this table gives you a pretty reasonable guess at the proper choice of the constant shift V_C. That is, you might set V_C as the one that minimizes potential mismatch to end layers. You can adjust If you start out with the choice V_C by invoking **lmpg** in an interactive mode, or by setting file **vshft** appropriately. If you do so, you will alleviate some of the burden on lmpg in determining this shift.

If the potential in either the L- or the R- differs significantly (see case L- and R- are both metals, below), or if there is significant deviation from charge neutrality in the end PL, the user should enlarge the embedding region, as the end PL are not sufficiently bulk-like. This difference should vanish at self-consistency if you construct the embedding region with PL near the end regions similar to those of the last region. Typically having 2 PL (say 0. . . 1, and n-2. . . n-1 does a pretty good job keeping the discrepancies between the bulk- and layer potentials small.

Now we must distinguish between metal and nonmetal cases. If either end layer is a nonmetal, its potential can shift by constant without affecting charge neutrality. Therefore, if either L- or R- regions is a nonmetal, there can be a constant shift in that region (change in band offset). Consider the following cases:

- L- and R- are both metals. No potential shifts are allowed in the end layers. In the self-consistency cycle (MODE=1) the program checks for deviations from charge neutrality, and adjusts the potential in the (C) region until neutrality is achieved. However, the density resulting from this Green's function will generate charges and electrostatic potentials V_m^i at sites i in the L- and R- layers. As mentioned above, potentials computed from the $(\ldots \,|\, \mathrm{L} \,|\, \mathrm{C} \,|\, \mathrm{R} \,|\, \ldots)$ system will reveal some differences relative to the electrostatic potentials V_b^i when computed using just the L- or R- charges and a geometry for infinitely repeating L- and R- layers. The potentials calculated these two ways is printed out, as described above.

- Only one of L- or R- is a metal (LMET=f and RMET=t or vise-versa). Now the nonmetallic end region can shift its potential by a constant to best align to the potential computed from the $(\ldots \,|\, \mathrm{L} \,|\, \mathrm{C} \,|\, \mathrm{R} \,|\, \ldots)$ system. We choose the constant in the nonmetallic PL that best aligns V_b^i and V_c^i; that is that minimizes the RMS difference V_b^i and V_c^i.

- Neither the L- or R- is a metal (LMET=f and RMET=f). If the C region is a metal (specified by BZ METAL=) the global potential shift should conform to the shift in the C region. Both endpoints must be allowed to float (there are two distinct band offsets). If no region is a metal, i.e. if there is no DOS at the Fermi level, the system should already be charge neutral and no shifts should be required.

1.4 Electrostatics in layer geometry

The correct procedure to construct electrostatics in a layer geometry is to carry out 2D Ewald summations for each PL, and add up the contributions from all PL. Because no one has come around to making 2D Ewald sums yet for this program, we use a trick.

We compute the electrostatics via an Ewald summation of the following supercell:

$$\overbrace{\mathrm{L-replica}}^{\mathrm{PLATL}} \mid \overbrace{\mathrm{L-replica}}^{\mathrm{PLATL}} \mid \overbrace{\mathrm{PL}\ 0 \mid \mathrm{PL}\ 1 \mid \mathrm{PL}\ 2 \mid \ldots \mid \mathrm{PL}\ n{-}1}^{\mathrm{PLAT}} \mid \overbrace{\mathrm{R-replica}}^{\mathrm{PLATR}} \mid \overbrace{\mathrm{R-replica}}^{\mathrm{PLATR}} \mid$$

It assumes periodic boundary conditions of period in the layer direction

```
PLAT(3) + 2*(PLATL + PLATR)
```

Note that this scheme is only approximate. It will eventually be replaced by 2D Ewald summations.

2 Input for the layer Green's function program

Most of the input for **lmpg** is similar to the band and crystal GF programs. This section describes input specific to **lmpg**.

Each site must be assigned a 'principal layer index' to tell **lmpg** which PL a site belongs to. In category SITE, each site should have a token PL:

```
ATOM='species-name' PL=layer-index
```

Each group of sites with the same PL index are grouped together into a single principal layer.
Remember that the principal layers should be large enough such that the range of the hamiltonian connects only adjacent PL. The range of the hamiltonian is fixed by the range of the structure; it is set in the STR category, token RMAX=
.
There is a **lmpg**-specific category, which includes the following:

```
PGF MODE=# PLATL=# # # PLATR=# # # GFOPTS= options SPARSE=#
```

Token MODE= tells **lmpg** what you want to do:

0 do nothing

1 calculate the diagonal GF, layer-by-layer. This is the appropriate mode for self-consistent calculations

2 left- and right-bulk bulk GF. The endpoints require special treatment, and this mode is designed to generate a self-consistent left- and right- bulk GF. It should be run before invoking **lmpg** with MODE=1.

3 find k(E) for left bulk. This uses a special trick (see PRB 39, 923 (1989)) to find the wave numbers of the left bulk GF corresponding to a given energy.

4 find k(E) for right bulk.

5 Calculate current using the Landauer formula.

Tokens specifying lattice vectors for the (... | L | C | R | ...) geometry:

Tokens	PLATL=# # #
and	PLATR=# # #
and	a re-definition of PLAT(1:3,3)

The reason why PLATL and PLATR must be specified is because **lmpg** must have information about the semi-infinite repeating layers that attach to each lead and extend to $\pm\infty$. Referring to the diagram below, there is implied a second L-replica of PL 0 shifted relative to PL 0 by -2*PLATL, another by -3*PLATL, and so on. Similarly there is implied a second R-replica shifted relative to PL n-1 by PLATR, another by 2*PLATR, and so on. This information is needed in order to make the crystalline Green's functions (and also surface Green's function) of each end layer.

$$\ldots\,|\,\overbrace{\mathrm{L-replica}}^{\mathrm{PLATL}}\,|\,\overbrace{\mathrm{PL}\ 0\,|\,\mathrm{PL}\ 1\,|\,\mathrm{PL}\ 2\,|\,\ldots\,|\,\mathrm{PL}\ n{-}1}^{\mathrm{PLAT}}\,|\,\overbrace{\mathrm{R-replica}}^{\mathrm{PLATR}}\,|\,\ldots$$

Caution: the inner product of PLATL with PLAT, and also the inner product PLATR with PLAT must be positive. That way each padding layer adds to the length of PLAT(3). (It is nonsensical for a principal layer to have a negative thickness.) Therefore, the program will stop if either dot product is negative.

Note: at present, **lmpg** calculates electrostatic potentials using a 3D supercell approach; see section 1.4. It does so by creating periodic boundary conditions in the third dimension with length

```
  PLAT(3) + 2*(PLATL + PLATR)
```

(Thus, PLATL and PLATR are needed in this second context as well, to carry out the Ewald summations). PLAT(1:3,3) printed as output by program **lmpg** reflects the addition of PLATL and PLATR.

Token GFOPTS=*options-list*

specifies a collection of optional extra functions. *options-list* is a series of strings string1;string2;... The following individual strings specify:

emom generate the output ASA moments, needed for self-consistency

idos make integrated properties, such as the sum of one-electron energies

dmat make the density-matrix $G_{RL,R'L'}$

sdmat make the site-diagonal density-matrix $G_{RL,RL'}$ The density matrix is written to a file 'dmat'

pdos Make the partial density of states (this has never been checked)

p3 Use third order potential functions

Token SPARSE=1

uses a modified LU decomposition to generate the layer GF. It tends to be significantly faster than the usual approach; compare test cases 5 and 6 in shell script pgf/test/test.pgf/

2.1 Potential shifts

File **vshft** holds information about potential shifts (sections 1.2 and 1.3) The global shifts are contained in the first line and keep information about Fermi level, the global constant shift, and the shifts at the L and R end regions needed to match the Fermi level. The syntax for the first line is

```
ef=# vconst=# vconst(L)=# vconst(R)=#
```

You can optionally add site-dependent potential shifts. After the first line, add a line `site shifts` followed by as many lines as desired, one line per site shift, e.g.:

```
ef=.03 vconst=-.01 vconst(L)=.02 vconst(R)=.03
site shifts
 3    .1
 4    .2
```

3 Program operation

lmpg starts by creating the left surface, and then proceeds 'left-to-right' layer-by-layer to generate the surface GF 0,1,2,3,... until the rightmost layer is reached. At that point the right surface GF is generated, and the crystal GF is generated by embedding between the left- and right- surface GF. Then using Dyson's equation, **lmpg** proceeds layer-by-layer 'right-to-left' to generate the crystal GF from the surface GF on the left and the crystal GF on the right. This is done for each energy and k-point in the two-dimensional BZ.

3.1 Use in conjunction with other programs

You can use plane-analysis lmplan to analyze charge distributions by plane, and also use it to create a 'site' file to facilitate making of lattices with periodic boundary conditions so that you can run programs 'lm' and 'lmgf' for comparison.

At present **lmpg** cannot make the static response function, as can **lmgf**; this can vastly improve effiency for self-consistency. However, if you make the response function using **lmgf**, you can use it with the **lmpg** program. Here is an example taken from the directory of pgf/tests/copt.

Invoke this command

```
   lmplan copt -vpgf=1 -cstrx=file -vlmf=f -vnk1=6 --pr31 -vnit=10
-vgamma=f -vsparse=0 --no-iactiv --time=5
```

At the prompt, type

```
wsite -pad abc
```

lmplan creates a site file named ‘abc.copt’ suitable for use with programs lm and **lmpg**. It uses the padding layers as buffer layers. Conveniently, it satisfies periodic boundary conditions. To verify this, try

```
> cp abc.copt site.copt
> lmchk copt -vpgf=0 -cstrx=file -vlmf=f -vnk1=6 --pr31 -vnit=10 -vgamma=f
-vsparse=0 --no-iactiv --time=5
```

Now you can make a suitable ASA static response function suitable for the layer code with

```
   > lmgf copt -vpgf=0 -cstrx=file -vlmf=f -vnk1=6 --pr31 -vnit=10
-vgamma=f -vsparse=0 --iactiv --time=5 -vscr=1
```

The *q=0* response function is written to file ‘psta.copt.’ (This file has already been created and sits in pgf/test/copt.) Once psta is created, it can greatly facilitate convergence to self-consistency. Try running, for example, the test script

```
> pgf/test/test.pgf --usepsta copt 5
```

will use the psta file to assist convergence when you use the switch –usepsta. Look in particular at the RMS DQ, viz:

```
> grep RMS out.lmpg.self-consistent
PQMIX: read 0 iter from file mixm.  RMS DQ=1.84e-3
PQMIX: read 1 iter from file mixm.  RMS DQ=1.59e-4 last it=1.84e-3
PQMIX: read 2 iter from file mixm.  RMS DQ=3.30e-4 last it=1.59e-4
PQMIX: read 3 iter from file mixm.  RMS DQ=4.66e-5 last it=3.30e-4
```

If you invoke it in the usual way, viz without –usepsta:

```
> pgf/test/test.pgf copt 5
```

you should see the following lines

```
> grep RMS out.lmpg.self-consistent
PQMIX: read 0 iter from file mixm.  RMS DQ=1.44e-3
PQMIX: read 1 iter from file mixm.  RMS DQ=6.41e-3 last it=1.44e-3
PQMIX: read 2 iter from file mixm.  RMS DQ=2.22e-1 last it=6.41e-3
PQMIX: read 3 iter from file mixm.  RMS DQ=3.05e-2 last it=2.22e-1
PQMIX: read 4 iter from file mixm.  RMS DQ=3.01e-2 last it=3.05e-2
PQMIX: read 5 iter from file mixm.  RMS DQ=2.66e-2 last it=3.01e-2
PQMIX: read 6 iter from file mixm.  RMS DQ=4.15e-2 last it=2.66e-2
PQMIX: read 7 iter from file mixm.  RMS DQ=4.92e-2 last it=4.15e-2
PQMIX: read 8 iter from file mixm.  RMS DQ=4.40e-2 last it=4.92e-2
PQMIX: read 8 iter from file mixm.  RMS DQ=2.24e-2 last it=4.40e-2
```

The improvement with the response function is dramatic.

Test cases

Shell script pgf/test/test.pgf checks out that the program is working properly, and it also is convenient to illustrate some of the features in **lmpg**.