Gmail Wiley Morgan <sholvah@gmail.com>

basis test

7 messages

Branton Campbell <branton@byu.edu> Mon, Feb 13, 2017 at 8:39 PM
To: Wiley Morgan <wiley.s.morgan@gmail.com>
Cc: "gus.hart@gmail.com" <gus.hart@gmail.com>

Hi Wiley,

Here's my conclusion from today's discussion.

Let P be the point matrix of any operator in lattice-space group G. Define the corresponding matrix S = 1-P, where 1 is the identity matrix. Though S is generally not invertible, some minimum matrix power n of S will be a diagonal matrix with only n's and 0's on the diagonal. Define S^-1 = (1/n)S^(n-1), which is not an actual inverse.

For every point operator P and integer triplet t, you want (H^-1).S.H.(S^-1).t to be another integer triplet. I "think" that this is the test you're looking for, though the idea needs inspection. We can talk again on Wednesday or later.

Branton

Wiley Morgan <wiley.s.morgan@gmail.com> Tue, Feb 14, 2017 at 9:29 AM
To: Branton Campbell <branton@byu.edu>

Dr. Campbell,

That would be great. Wednesday after 11 AM or Friday after 2 PM would work best for me. What would work for you?

Wiley Morgan
https://github.com/wsmorgan
[Quoted text hidden]

Branton Campbell <branton@byu.edu> Tue, Feb 14, 2017 at 6:12 PM
To: Wiley Morgan <wiley.s.morgan@gmail.com>
Cc: "gus.hart@gmail.com" <gus.hart@gmail.com>

Wiley,

I have time between 2 and 3 tomorrow. Here's a bit more detail.

The position p and the translational-component t of a symmorphic-space-group operator (P,t) are related. Note that p is the non-variable part of the operator position, and that t is a translation perpendicular to the line or plane of the operator. The operator transforms coordinates as x' = P.x+t, where x' is also equal to P(x-p)+p [slide coordinate system so at to move operator to origin, apply point operation, and slide operator back to its original position]. Hence t = (1-P)p or t = S.p where S = 1-P. This worked well on a few cases that I tried by hand. It looks valid though I haven't checked any books.

Determine the minimal matrix power n that takes S to a diagonal form with only zeros and n's on the diagonal, and define S^-1 = (1/n)S^(n-1), which is not a true matrix inverse when S is not invertible. Then p = (S^-1).t.

You want the parent and subgroup to be isomorphic, so that there is a bijective mapping between the parent subgroup operators. In the subgroup, the operators get more spread out spatially but have the same overall pattern. If H is the HNF column matrix that defines the supercell basis relative to the parent, the position of a given operator in the parent is easily mapped (actually moved) to a corresponding position p* in the supercell (but still in parent-cell coordinates) as follows: p* = H.p.

Then the translational component t* of the operator at p* is calculated as t* = S.p* = S.H.p = S.H.(S^-1).t.

And in the setting of the supercell, this translation is described as t*’ = (H^-1).t*, so that t*’ =M.t where M = (H^-1).S.H.(S^-1). The matrix M looks a lot like a commutator, but not truly since S is not generally invertible.

In any case, t can be any integer triplet since the parent group G is symmorphic. And the resulting t*’ must be an integer triplet since the subgroup G’ is also symmorphic and since t*’ is presented in the setting of the subgroup. Thus M must transform every integer triplet into an integer triplet for every pointe operator P of G. This provides a system of integer equations in the elements of H. You want the set of all integer solutions to these equations.

Best wishes,

Branton

From: sholvah@gmail.com [mailto:sholvah@gmail.com] **On Behalf Of** Wiley Morgan
Sent: Tuesday, February 14, 2017 9:30 AM
To: Branton Campbell <branton@byu.edu>
Subject: Re: basis test

[Quoted text hidden]

Branton Campbell <branton@byu.edu> Wed, Feb 15, 2017 at 8:24 PM
To: Wiley Morgan <wiley.s.morgan@gmail.com>
Cc: "gus.hart@gmail.com" <gus.hart@gmail.com>

Hi Wiley,

Here’s an improved version of the previous message based on the discussions today.

A space group operator consists of a point operation P and a translation t. For the 73 symmorphic space groups, which are semi-direct products of a translation group T and a point group P, the translational components are always integer triplets. You are interested only in the 14 lattice space groups, which are all symmorphic.

The translation component t of operator (P,t) is the sum of invariant and variant parts $t = t_i$ and t_v. The projection operator Q of P is calculated as $Q = (1/n)\mathrm{Sum}[P^m,\{m,1,n\}]$ where n is the matrix order of P. Then $t_i = Q.t$ and $t_v = (1-Q).t$. Similarly, if p is a coordinate point on the symmetry operator, then $p_i = Q.p$ and $p_v = (1-Q).p$. We know that p_v is the shortest vector pointing from the origin to the symmetry element, and will always be perpendicular to the line or plane of the element for rotation (or screw) axes and mirror (or glide) planes. In the discussion that follows, the translation components t and positions p will implicitly refer to their variant components.

The position and the translational-component t of a symmorphic-space-group operator (P,t) are related. The operator transforms coordinates as $x' = P.x+t$, where x' is also equal to $P(x-p)+p$ [slide coordinate system so as to move the operator to origin, apply point operation, and slide the coordinate system so as to move the operator back to its original position]. Hence $t = (1-P)p$ or $t = S.p$ where $S = 1-P$.

Though S is generally not invertible, one can always find a matrix that we'll call S^{-1} (not an actual matrix inverse) such that $p = (S^{-1}).t$. Where there is flexibility, choose the simplest matrix that works (zeros are best, small positive integers are next best, then simple fractions, etc). Note that if you compile a list of all of the point operators from all of the lattice space groups, you only have to find the appropriate S and S^{-1} once for each one; then save them for future use.

You want the parent and subgroup to be isomorphic, which requires a linear mapping between the operators of the parent and the operators of the subgroup. In the subgroup, the operators get more spread out spatially but have the same overall pattern. Let B be the column matrix that defines the supercell basis relative to the parent. In general, B can be decomposed as $U.H$ where U is unimodular and H is in Hermite normal form. You specifically indicated that you are only interested in cases where $U = 1$; but we still need to make sure that U can be accommodated within the theory. The position of a given operator in the parent can be mapped (actually moved) to a corresponding position p^* in the supercell (but still represented in parent-cell coordinates) as follows: $p^* = H.p$ (not $B.p$). We need to examine this assertion, though it seems clear that $U.H$ should be viewed as a two-stage mapping, where H must move operators and where U must not. For example, if H is the identity matrix, U merely switches to an equivalent setting G' of the parent space group, which is guaranteed to be isomorphic to G; the only non-translational linear mapping that moves all of the operators of G (and also the asymmetric cell of G) to those of G' is the identity.

Then the translational component t^* of the operator at p^* is calculated as $t^* = S.p^* = S.H.p = S.H.(S^{-1}).t$.

And in the setting of the supercell, this translation is described as $t^{*\prime} = (B^{-1}).t^*$, so that $t^{*\prime} = M.t$ where $M = (U^{-1}).(H^{-1}).S.H.(S^{-1})$.

In any case, t can be any integer triplet since the parent group G is symmorphic. And the resulting $t^{*\prime}$ must be an integer triplet since the subgroup G' is also symmorphic and since $t^{*\prime}$ is presented in the setting of the subgroup. Thus M must transform every integer triplet into an integer triplet for every point operator P of G. This provides a system of integer equations in the elements of H, which you must combine with the integer inequalities that define the form of H. You then want to extract the value of H from each integer solution to this system of equations and inequalities. Mathematica's "Reduce" command will provide a finite list of infinite families of solutions in compact form.

You only need to do this once for each lattice space group (in Mathematica), which shouldn't take long. The compact form of each family of solutions could then be coded up manually in Fortran.

Best wishes,

Branton

From: sholvah@gmail.com [mailto:sholvah@gmail.com] **On Behalf Of** Wiley Morgan
Sent: Tuesday, February 14, 2017 9:30 AM
To: Branton Campbell <branton@byu.edu>
Subject: Re: basis test

Dr. Campbell,

[Quoted text hidden]
[Quoted text hidden]

Branton Campbell <branton@byu.edu> Tue, Feb 21, 2017 at 7:43 PM
To: Wiley Morgan <wiley.s.morgan@gmail.com>
Cc: "gus.hart@gmail.com" <gus.hart@gmail.com>

Hi Wiley,

When we met last, we discussed a simpler way to think about the problem that involved only the translational components (not the positions) of the space-group operators. I summarize the new approach.

A space group operator consists of a point operation P and a translation t. For the 73 symmorphic space groups, which are semi-direct products of a translation group T and a point group P, the translational components are always integer triplets. You are interested only in the 14 lattice space groups, which are all symmorphic.

You want the parent group G and the subgroup to be isomorphic, which requires a linear mapping between the operators of the parent and the operators of the subgroup. In the subgroup, the operators get more spread out spatially but have the same overall pattern. Let B be the column matrix that defines the supercell basis relative to the parent. In general, B can be decomposed as U.H where U is unimodular and H is in Hermite normal form, though you are only interested in H.

Let each operator (P,t) of the parent be similarity-transformed to the equivalent operator (P',t') in the setting of the supercell so that P' = (H^-1).P.H and t' = (H^-1).t. This is a linear (homomorphic) and bijective mapping. Those operators of the parent that are also in the subgroup must have integer translations in the primitive setting of the subgroup (by definition). So we define the set S to be the collection of operators (P', t') with integer translations and discard all others. However, the set S is not generally a subgroup of the parent; the multiplication of its operators is not closed when any of the P' contain non-integer matrix elements. In fact, the following three statements are equivalent.

(A) The set S is a subgroup of the parent group (i.e. closed under multiplication).

(B) The set S is isomorphic to the parent group.

(C) The point parts P' of the set S contain only integers.

This equivalence is obvious in the light of two additional facts.

Fact #1: No matter the H matrix employed, every point operator P of the parent will be mapped to at least one operator (P', t') that has integer translations. Thus when we collect all of the operators (P', t') with integer translations and discard the others, every point operator of the parent will be represented (i.e. the point groups of the parent and the set S are identical).

Fact #2: The product of two operators in S is (P3', t3') = (P2', t2').(P1', t1') = (P2'.P1', P2'.t1' + t2'). It's obvious that if t1' and t2' are all-integer, then t3' is all-integer if and only if P2'.t1' is all-integer, which is true if and only iff P2' is an all-integer matrix.

Summary: to test H to see if it results in a subgroup S isomorphic to G, you need only check that the transformed point operators are all-integer.

Best,

Branton

From: Branton Campbell
Sent: Wednesday, February 15, 2017 8:24 PM
To: 'Wiley Morgan' <wiley.s.morgan@gmail.com>
Cc: gus.hart@gmail.com
Subject: RE: basis test

Hi Wiley,

Here's an improved version of the previous message based on the discussions today.

A space group operator consists of a point operation P and a translation t. For the 73 symmorphic space groups, which are semi-direct products of a translation group T and a point group P, the translational components are always integer triplets. You are interested only in the 14 lattice space groups, which are all symmorphic.

The translation component t of operator (P,t) is the sum of invariant and variant parts t = t_i and t_v. The projection operator Q of P is calculated as Q = (1/n)Sum[P^m,{m,1,n}] where n is the matrix order of P. Then t_i = Q.t and t_v = (1-Q).t. Similarly, if p is a coordinate point on the symmetry operator, then p_i = Q.p and p_v = (1-Q).p. We know that p_v is the shortest vector pointing from the origin to the symmetry element, and will always be perpendicular to the line or plane of the element for rotation (or screw) axes and mirror (or glide) planes. In the discussion that follows, the translation components t and positions p will implicitly refer to their variant components.

The position and the translational-component t of a symmorphic-space-group operator (P,t) are related. The operator transforms coordinates as x' = P.x+t, where x' is also equal to P(x-p)+p [slide coordinate system so as to move the operator to origin, apply point operation, and slide the coordinate system so as to move the operator back to its original position]. Hence t = (1-P)p or t = S.p where S = 1-P.

Though S is generally not invertible, one can always find a matrix that we'll call S^-1 (not an actual matrix inverse) such that p = (S^-1).t. Where there is flexibility, choose the simplest matrix that works (zeros are best, small positive integers are next best, then simple fractions, etc). Note that if you compile a list of all of the point operators from all of the lattice space groups, you only have to find the appropriate S and S^-1 once for each one; then save them for future use.

You want the parent and subgroup to be isomorphic, which requires a linear mapping between the operators of the parent and the operators of the subgroup. In the subgroup, the operators get more spread out spatially but have the same overall pattern. Let B be the column matrix that defines the supercell basis relative to the parent. In general, B can be decomposed as U.H where U is unimodular and H is in Hermite normal form. You specifically indicated that you are only interested in cases where U = 1; but we still need to make sure that U can be accommodated within the theory. The position of a given operator in the parent can be mapped (actually moved) to a corresponding position p* in the supercell (but still represented in parent-cell coordinates) as follows: p* = H.p (not B.p). We need to examine this assertion, though it seems clear that U.H should be viewed as a two-stage mapping, where H must move operators and where U must not. For example, if H is the identity matrix, U merely switches to an equivalent setting G' of the parent space group, which is guaranteed to be isomorphic to G; the only non-translational linear mapping that moves all of the operators of G (and also the asymmetric cell of G) to those of G' is the identity.

Then the translational component t* of the operator at p* is calculated as t* = S.p* = S.H.p = S.H.(S^-1).t.

And in the setting of the supercell, this translation is described as t*' = (B^-1).t*, so that t*' =M.t where M = (U^-1).(H^-1).S.H.(S^-1).

In any case, t can be any integer triplet since the parent group G is symmorphic. And the resulting t*' must be an integer triplet since the subgroup G' is also symmorphic and since t*' is presented in the setting of the subgroup. Thus M must transform every integer triplet into an integer triplet for every point operator P of G. This provides a system of integer equations in the elements of H, which you must combine with the integer inequalities that define the form of H. You then want to extract the value of H from each integer solution to this system of equations and inequalities. Mathematica's "Reduce" command will provide a finite list of infinite families of solutions in compact form.

You only need to do this once for each lattice space group (in Mathematica), which shouldn't take long. The compact form of each family of solutions could then be coded up manually in Fortran.

Best wishes,

Branton

From: sholvah@gmail.com [mailto:sholvah@gmail.com] **On Behalf Of** Wiley Morgan
Sent: Tuesday, February 14, 2017 9:30 AM
To: Branton Campbell <branton@byu.edu>
Subject: Re: basis test

Dr. Campbell,

[Quoted text hidden]
[Quoted text hidden]

Branton Campbell <branton@byu.edu> Tue, Feb 21, 2017 at 7:48 PM
To: Wiley Morgan <wiley.s.morgan@gmail.com>, "gus.hart@gmail.com" <gus.hart@gmail.com>

Gus and Wiley,

With the Mathematica code that I wrote two weeks ago to solve systems of integer inequalities (for the normal-subgroup problem), I think that we could solve your HNF problem once and for all for every possible case in about an hour. We should get together sometime to discuss this.

Branton

Wiley Morgan <wiley.s.morgan@gmail.com> Thu, Feb 23, 2017 at 2:32 PM
To: Branton Campbell <branton@byu.edu>
Cc: Wiley Morgan <wiley.s.morgan@gmail.com>, "gus.hart@gmail.com" <gus.hart@gmail.com>

Dr. Campbell,

I've been trying to test out the ideas we've talked about and have some questions about implementation. However, if you think that your Mathematica notebook can find a solution for us then we should try it. Dr. Hart will be tied up for the next week or so since he's going to a conference next week but I'm free to meet with you when you have some time to try it.

Wiley Morgan
https://github.com/wsmorgan
[Quoted text hidden]