

The Rules of Physics

J.A.M. Vermaseren

- Motivation, Part 1
- Motivation, Part 2
- The derivation of reduction equations
- The rules of GO
- Dealing with go problems
- Similarities
- The rules of Physics

Motivation, Part 1

This talk is motivated by the calculation of three loop structure functions in DIS. This is a gigantic project that has already consumed more than 20 manyears. In addition it has led to many new techniques in the computation of loop integrals.

The main body of these calculations is the evaluation of all Mellin moments of the DIS structure functions as a function of the moment number N . To be able to do this one has to derive many formulas that enable the systematic reduction of all integrals that can be encountered into either simpler integrals, integrals that can be done directly, or integrals that can be computed from difference equations in which the inhomogeneous term consists of simpler integrals. The main work is the derivation of these equations. Of course, running the computer programs that use these equations is a major job as well as there are many years of CPU power involved. However, if the programs are sufficiently generic, the running is mainly the organization of sufficient amounts of computer power.

Another major part of the work is of course making sure that there are no significant errors. This is a science by itself, but we will not address that in this context.

The derivation of the necessary equations ($O(1000)$ in the case of this calculation) is done by computer as well.

- For each topology one can write down a number of equations based on integration by parts and other symmetry and invariance principles.
- These equations then have to be combined to give useful equations.
- The problem is they are parametric equations and one may have to shift the values of these parameters to make useful combinations.
- Then, some combinations possess nasty properties and don't suffice. The whole can only be done by guessing combinations and developing insight.

The last consideration is similar to strategy games like chess or go. For reasons that should become clear we will compare here with go.

The main motivation of this talk is to stimulate crossfertilization between perturbative field theory and the writing of programs for strategic games like go.

Motivation, Part 2

Before actually starting a calculation there are several considerations to be made:

- With current technology, when will I finish?
- What are the prospects of obtaining better methods?
- How fast can one obtain better methods?
- With these better methods, when will I finish?

It should be clear that if the finishing date with current technology is way after the need for the results one will be forced to look for better methods. But it should be equally clear that looking for better methods is so time consuming that it could push the 'improved finishing time' way beyond the 'current technology finishing time'. In that case we should just start. An example of the latter is working for one day to make a program run in 10 minutes that before would run in 1 hour. That should be wasted effort, unless the program is to be used many times. But we will not consider that case here.

The usual case is that initial improvements will bring the finishing time closer but after a while one has reached the nearest finishing time and further search for improvement will only make the finishing time later. Unfortunately we do not know when this occurs. We have to guess.

Motivation, Part 2

Before actually starting a calculation there are several considerations to be made:

- ◉ With current technology, when will I finish?
- What are the prospects of obtaining better methods?
- How fast can one obtain better methods?
- ◉ With these better methods, when will I finish?

It should be clear that if the finishing date with current technology is way after the need for the results one will be forced to look for better methods. But it should be equally clear that looking for better methods is so time consuming that it could push the 'improved finishing time' way beyond the 'current technology finishing time'. In that case we should just start. An example of the latter is working for one day to make a program run in 10 minutes that before would run in 1 hour. That should be wasted effort, unless the program is to be used many times. But we will not consider that case here.

The usual case is that initial improvements will bring the finishing time closer but after a while one has reached the nearest finishing time and further search for improvement will only make the finishing time later. Unfortunately we do not know when this occurs. We have to guess.

An additional problem is that experimentalists need our results at a given time. This suggests the optimal way to proceed: Once the finishing time is well before when experimentalists need the results and once obvious improvements are not foreseeable, one group should start the calculation, while other groups could possibly try to make improvements. In this talk we will be addressing the plight of the group that starts the calculation, even though it may still be a bit premature. One can always try to improve methods when the other groups find extra technology.

Hence, the remark: "If you need thousands of hours of CPU time and many years of people time, you should try to be smarter" does not hold. It is comparable to the remark: "If the answer to your integral is one, there must be a simple way to get it".

The derivation of reduction equations

The way the current calculations are done is by reduction equations:

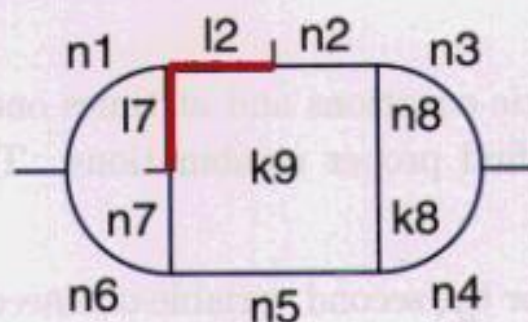
- One writes down all equations that can be thought of for a given system.
- One tries to combine these equations in such a way that step by step parameters are either brought to a value that brings one to a simpler case or brings the parameter to a standard value like 1 or 0.
- In the end one has either only integrals that are of a simpler type, or one has a master integral.
- In our case the master integral is then determined by a difference equation. We have encountered even a fourth order equation. This equation has in its inhomogeneous part only simpler integrals. We have programs to automatically solve these difference equations.

One might wonder: “what is so special about combining linear equations?” The answer is that there are a few complications:

- These are parametric equations and at times one needs to shift the parameters to find proper combinations. There are up to $12 + 1$ variables.
- To find equations for the second variable one needs to substitute the effects of the first reduction equation in all other equations. After a few steps the remaining equations tend to become rather lengthy. In our reductions we have many reduction identities with thousands of terms.
- There is always the possibility of spurious poles. These are powers of $1/\epsilon$ that could be avoided if one combines the equations in a more careful manner. Sometimes they occur only for particular values of the remaining parameters (including N).
- The resulting reduction scheme should be executable. With executable we mean: inside an available amount of CPU time and inside available memory.

Of course this last condition might make the whole thing impossible, but let us assume for now that it means: reasonably close to a fastest and most concise solution.

Let us have a look at an example. This concerns a subtopology of the ladder topology. The diagram is:



An example of an equation that can be used to reduce n_4 (in ladder notation) is

$$\begin{aligned}
 &+LA27(17, -1+l2, n1, n2, n3, n4, n5, n6, n7, n8, 0, 0, -1+k8, k9)*(-k8) \\
 &+LA27(17, l2, -1+n1, n2, n3, n4, n5, n6, n7, n8, 0, 0, k8, -1+k9)*(k9) \\
 &+LA27(17, l2, n1, -1+n2, n3, n4, n5, n6, n7, 1+n8, 0, 0, k8, k9)*(n8) \\
 &+LA27(17, l2, n1, -1+n2, n3, n4, n5, n6, n7, n8, 0, 0, -1+k8, k9)*(k8) \\
 &+LA27(17, l2, n1, -1+n2, n3, n4, n5, n6, n7, n8, 0, 0, k8, -1+k9)*(-k9) \\
 &+LA27(17, l2, n1, n2, -1+n3, 1+n4, n5, n6, n7, n8, 0, 0, k8, k9)*(-n4) \\
 &+LA27(17, l2, n1, n2, -1+n3, n4, n5, n6, n7, 1+n8, 0, 0, k8, k9)*(-n8) \\
 &+LA27(17, l2, n1, n2, n3, 1+n4, n5, n6, n7, n8, 0, 0, k8, k9)*Q.Q*(n4) \\
 &+LA27(17, l2, n1, n2, n3, n4, n5, n6, -1+n7, n8, 0, 0, k8, -1+k9)*(-k9) \\
 &+LA27(17, l2, n1, n2, n3, n4, n5, n6, n7, n8, 0, 0, k8, k9)* \\
 &\quad (4+k9+k8-n8-n4-2*n3-2*ep);
 \end{aligned}$$

This equation is rather uncomplicated and is usually one of the first to be used. It allows us to bring n_4 down to 1, sometimes at the cost of raising some other parameters. We will rewrite it as

```

id LA27(17?pos_,12?pos_,n1?pos_,n2?,n3?pos_,n4?{>1},n5?,n6?pos_,
        ,n7?pos_,n8?pos_,0,0,k8?,k9?) = -1/Q.Q/(-1+n4)*
+LA27(17,12,n1,n2,-1+n3,n4,n5,n6,n7,n8,0,0,k8,k9)*(-(-1+n4))
+LA27(17,-1+12,n1,n2,n3,-1+n4,n5,n6,n7,n8,0,0,-1+k8,k9)*(-k8)
+LA27(17,12,-1+n1,n2,n3,-1+n4,n5,n6,n7,n8,0,0,k8,-1+k9)*(k9)
+LA27(17,12,n1,-1+n2,n3,-1+n4,n5,n6,n7,1+n8,0,0,k8,k9)*(n8)
+LA27(17,12,n1,-1+n2,n3,-1+n4,n5,n6,n7,n8,0,0,-1+k8,k9)*(k8)
+LA27(17,12,n1,-1+n2,n3,-1+n4,n5,n6,n7,n8,0,0,k8,-1+k9)*(-k9)
+LA27(17,12,n1,n2,-1+n3,-1+n4,n5,n6,n7,1+n8,0,0,k8,k9)*(-n8)
+LA27(17,12,n1,n2,n3,-1+n4,n5,n6,-1+n7,n8,0,0,k8,-1+k9)*(-k9)
+LA27(17,12,n1,n2,n3,-1+n4,n5,n6,n7,n8,0,0,k8,k9)*
(5-2*ep-2*n3-n4-n8+k8+k9));

```

Both present us with problems. The first equation may follow from zero to one. This could lead us to a case of asymptotic behavior. The reason is that integrals with all lines present we need to know to order 1 (and we will tabulate them only to that precision) while integrals in which the five-line is missing might have a better behavior. Hence we would need the corresponding integral in which we have again to order ϵ .

The next example is already nastier. We assume that in a similar way we have brought n_3 down to 1. Next we look at n_6 for which there are two possibilities:

$$\begin{aligned}
 &+LA27(1+17,12,n1,n2,1,1,n5,n6,-1+n7,n8,0,0,k8,k9)*(-17) \\
 &+LA27(17,12,1+n1,-1+n2,1,1,n5,n6,n7,n8,0,0,k8,k9)*(n1) \\
 &+LA27(17,12,1+n1,n2,1,1,n5,n6,-1+n7,n8,0,0,k8,k9)*(-n1) \\
 &+LA27(17,12,n1,n2,1,1,-1+n5,1+n6,n7,n8,0,0,k8,k9)*(n6) \\
 &+LA27(17,12,n1,n2,1,1,n5,1+n6,-1+n7,n8,0,0,k8,k9)*(-n6) \\
 &+LA27(17,12,n1,n2,1,1,n5,n6,n7,n8,0,0,k8,k9)* \\
 &\quad (4+k9-2*n7-n6-n1-17-2*ep);
 \end{aligned}$$

and

$$\begin{aligned}
 &+LA27(1+17,-1+12,n1,n2,1,1,n5,n6,n7,n8,0,0,k8,k9)*(17) \\
 &+LA27(1+17,12,-1+n1,n2,1,1,n5,n6,n7,n8,0,0,k8,k9)*(-17) \\
 &+LA27(17,12,-1+n1,n2,1,1,n5,1+n6,n7,n8,0,0,k8,k9)*(-n6) \\
 &+LA27(17,12,-1+n1,n2,1,1,n5,n6,1+n7,n8,0,0,k8,k9)*(-n7) \\
 &+LA27(17,12,n1,-1+n2,1,1,n5,n6,1+n7,n8,0,0,k8,k9)*(n7) \\
 &+LA27(17,12,n1,-1+n2,1,1,n5,n6,n7,n8,0,0,k8,-1+k9)*(-k9) \\
 &+LA27(17,12,n1,n2,0,1,n5,n6,n7,n8,0,0,k8,-1+k9)*(k9) \\
 &+LA27(17,12,n1,n2,1,1,n5,1+n6,n7,n8,0,0,k8,k9)*Q.Q*(n6) \\
 &+LA27(17,12,n1,n2,1,1,n5,n6,n7,-1+n8,0,0,k8,-1+k9)*(-k9) \\
 &+LA27(17,12,n1,n2,1,1,n5,n6,n7,n8,0,0,k8,k9)* \\
 &\quad (4+k9-n7-n6-2*n1-17-2*ep);
 \end{aligned}$$

Both present us with problems. The first equation may raise n_5 from zero to one. This could lead us to a case of a spurious pole. The reason is that integrals with all lines present we need to know only to order 1 (and we will tabulate them only to that precision), while integrals in which the five-line is missing might have a factor $1/\epsilon$ and hence we would need the corresponding integral in which n_5 is one again to order ϵ).

The second equation does not have this problem but it may raise n_7 . This can lead to extremely complicated integrals for which we may not have enough computer resources. Hence the solution is to use the first equation for the derivation of the next equations and to implement both in the reduction equations. We will use the first equation if the spurious pole doesn't do any harm. Otherwise we will have to resort to the second equation.

There is one equation that is relatively simple. It should be used like a joker in the sense that one should keep it for the right moment. Indiscriminate use at the wrong moment may complicate matters:

$$\begin{aligned} &+LA27(1+17,12,n1,n2,n3,n4,n5,n6,-1+n7,n8,0,0,k8,k9)*(-17) \\ &+LA27(17,1+12,n1,-1+n2,n3,n4,n5,n6,n7,n8,0,0,k8,k9)*(-12) \\ &+LA27(17,12,n1,n2,n3,n4,n5,n6,n7,n8,0,0,k8,k9)* \\ &\quad (-k8+17+12+N); \end{aligned}$$

The presence of the factor N in the rhs causes a slight complication if there are extra powers of $P \cdot Q$:

$$\begin{aligned} \text{id LA27}(17?pos_,12?\{>1\},n1?pos_,n2?,1,1,n5?,n6?pos_, \\ \quad ,n7?pos_,n8?pos_,0,0,k8?,k9?)*P.Q^n? = P.Q^n/(-1+12)*(\\ +LA27(1+17,-1+12,n1,1+n2,1,1,n5,n6,-1+n7,n8,0,0,k8,k9)*(-17) \\ +LA27(17,-1+12,n1,1+n2,1,1,n5,n6,n7,n8,0,0,k8,k9)* \\ \quad (N-n-1+12+17-k8)); \end{aligned}$$

Note that each power of $P \cdot Q$ lowers the effective value of N .

The final equation in the reduction of this case is a second order difference equation that has 38 different simpler integrals in its inhomogeneous term.

The rules of go

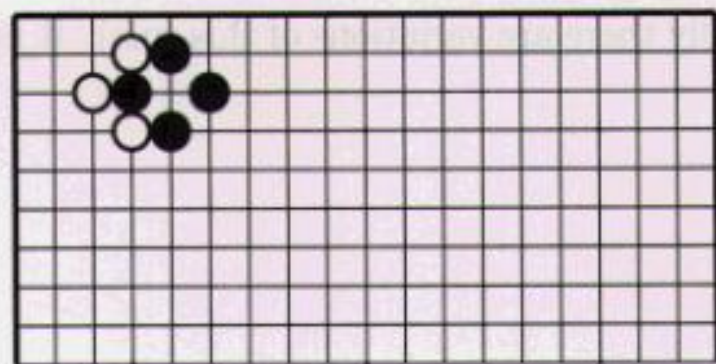
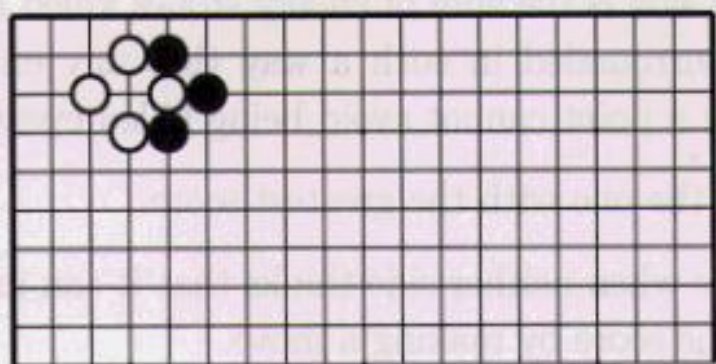
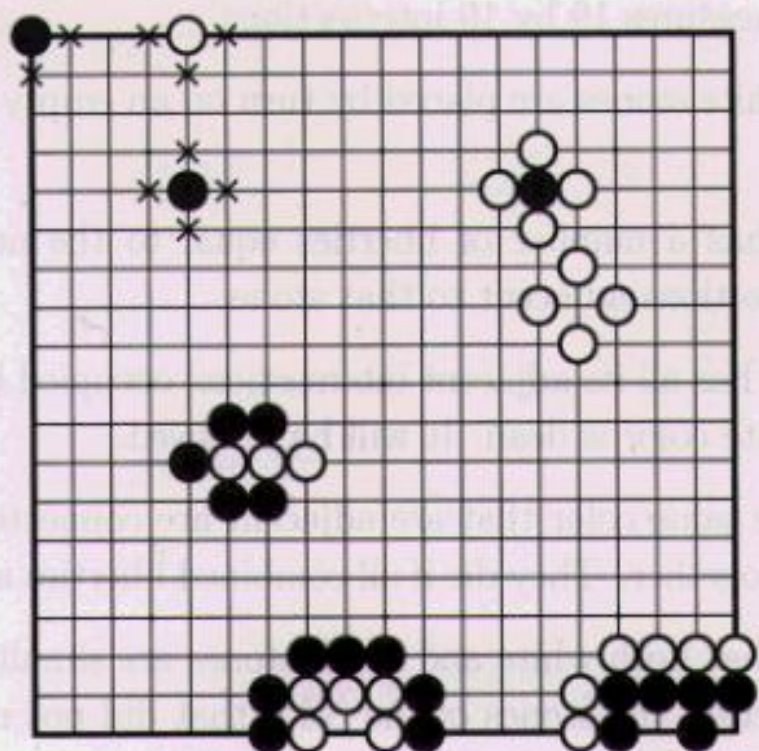
I have selected go here as the game of choice because of a variety of reasons:

- We are in Japan.
- I am most familiar with it.
- The 'standard' brute force programming techniques don't work here, unlike chess.

Maybe not everybody is familiar with the rules of go, so here they are in a nutshell:

- The board measures 19 by 19 intersections
- Black and white stones are placed by turn on an empty intersection.
- Each stone has a number of liberties equal to the number of empty intersections adjacent to that stone.
- A stone that has all its adjacent intersections occupied by stones of the opposite color is dead. It will be removed.
- Stones of the same color that are adjacent are connected. They live and die together. They die if all combined liberties are taken.
- In the case that both white and black stones are simultaneously without liberties, the stones of the color that did not move last are removed.
- The score of a side is the sum of enemy stones killed and empty intersections surrounded in such a way that any enemy stone placed on such a point cannot avoid being killed eventually.
- The winner is the one with the greatest score.
- The game ends when neither side thinks that it can improve the difference in the score by making a move.
- Repetition of a position with the same color at move is not allowed. (Actually there are variations of this rule). It is called the KO rule.





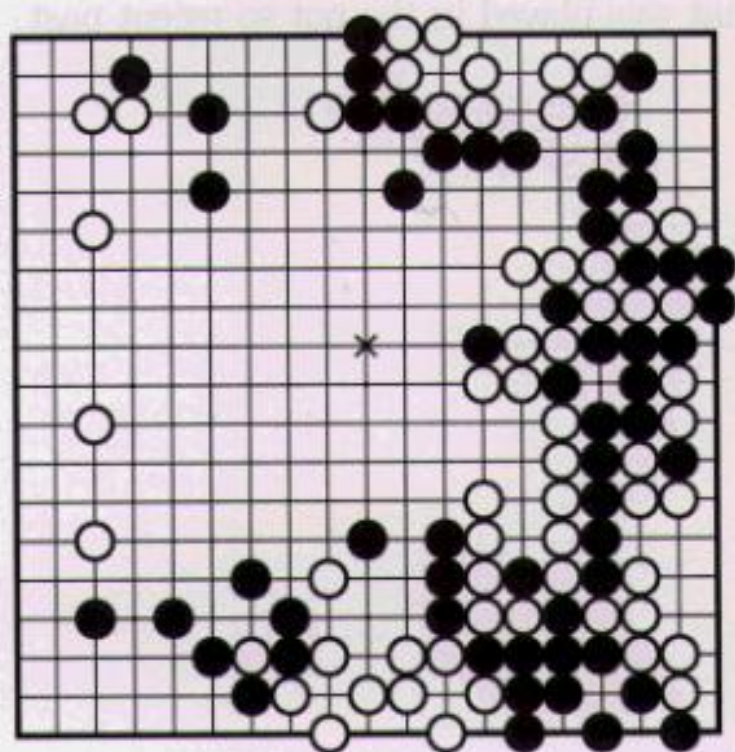
Dealing with go problems

We will have a look now at what type of thinking is involved in playing go at a decent level. The first and most complicated example is from a game that was played in the not so recent past.



- It gives some support to the lower middle black group.
 - It restricts the white group on the middle right.
 - It helps expanding the black influence at the top.
 - It prepares the invasion of the left.
- Of course, this is all backed up by a very accurate look ahead.

The following is a situation from a game of go. A rather famous game and any very strong go player would recognize it for its next move is one of the most famous in go history. The game was played in 1846. White was Gennan Inseki and black the young upcoming star Shusaku.

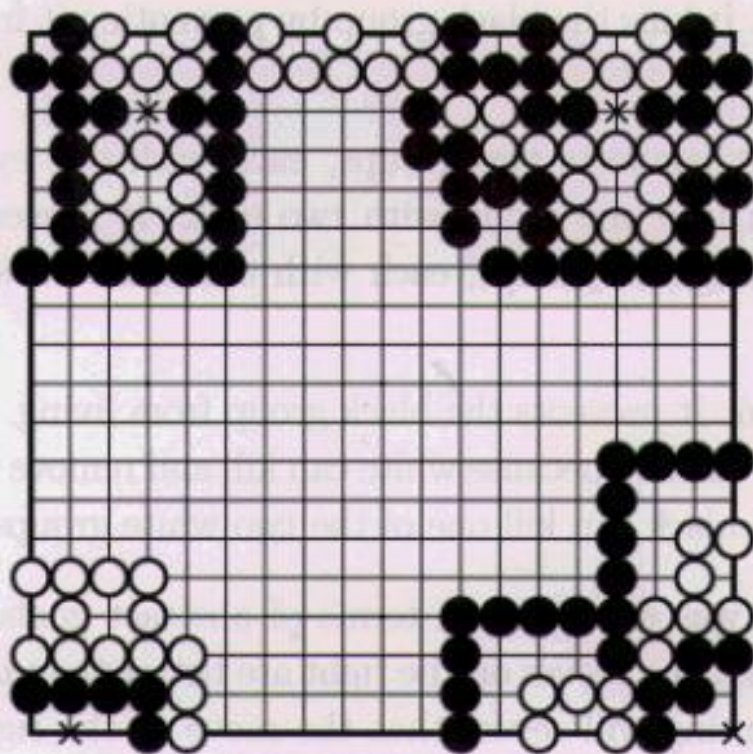


The reason why this game is interesting for us is the thinking behind the next move, once one oversimplifies a bit. It solves 4 problems in the various positions at the same time:

- It gives some support to the lower middle black group.
- It restricts the white group on the middle right.
- It helps expanding the black influence at the top.
- It prepares the invasion of the left.

Of course this is all backed up by a very accurate look ahead.

For us the style of thinking here is important: it is project oriented thinking. Let us look at some simpler examples of this:

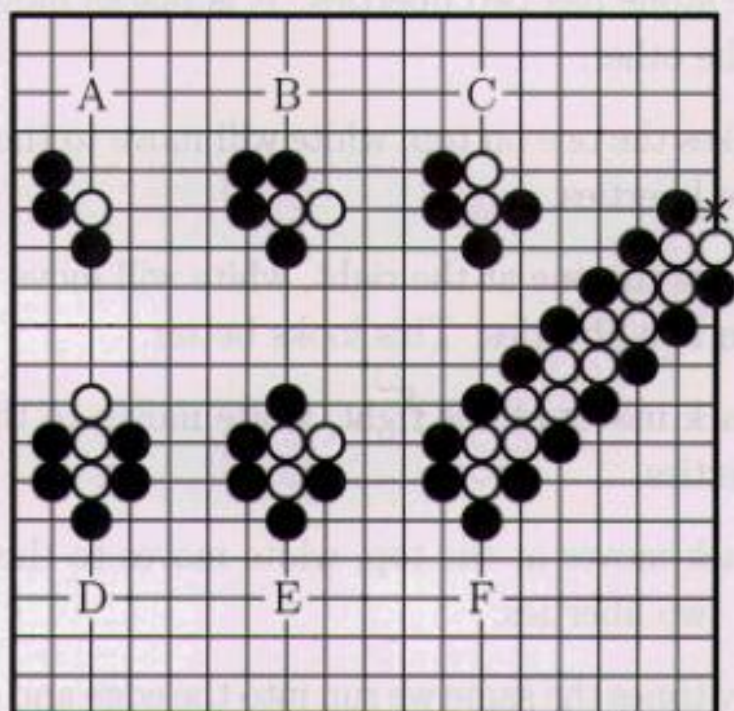


Here we see four separate situations. Let us assume it is whites move. The moves under consideration are indicated with crosses. What do these moves do?

- Left-Top: It connects two groups that would not be separately alive, but together they are.
- Left-Bottom: It kills the black group by preventing it from making two eyes.
- Right-Top: It connects two groups, each with one eye, hence making them into one group with two eyes. It prevents black from connecting two groups, each with one eye. These groups will die now.
- Right-Bottom: It prevents the black group from living. Now the white groups can live because white can kill and remove the black stones before black can kill one of the two white groups.

Notice that all was expressed in terms of abstract goals. And it should be clear that the moves on the right are better than the moves on the left. Counting will reveal that the move at the right top is worth 72 points and the move at the right bottom 50 points.

Another simple example. The shicho or ladder.



We see here a problem in which black would like to capture white.

- A: The white stone has two liberties. It is black's move. He can take one or the other.
- B: If black takes the one on top, white will move to the right and has now three liberties.
- C: If black takes the one at the right, white will move to the top and has again two liberties. This looks better.
- D: If next black moves at the right, white moves to the top and has three liberties.
- E: If next black moves at the top, white moves to the right and has still only two liberties.
- F: After many times the same we run into the edge and eventually white will end up with only one liberty. In the next move black can kill all at the cross.

Things can become more complicated when other white or black stones are encountered. Because the encirclement is very fragile, one must work out these ladders before they are started. Almost never they are played.

But again this is a good example of project oriented thinking.

- A: The white stone has two liberties. It is black's move. He can take one or the other.
- B: If black takes the one on top, white will move to the right and has now three liberties.
- C: If black takes the one at the right, white will move to the top and has again two liberties. This looks better.
- D: If next black moves at the right, white moves to the top and has three liberties.
- E: If next black moves at the top, white moves to the right and has still only two liberties.
- F: After many times the same we run into the edge and eventually white will end up with only one liberty. In the next move black can kill all at the cross.

Things can become more complicated when other white or black stones are encountered. Because the encirclement is very fragile, one must work out these ladders before they are started. Almost never they are played.

But again this is a good example of project oriented thinking.

Of course in reality things are more vague, but by the time the endgame nears things become a little bit clearer. The abstractions one has to think about in the opening are rather different from those in the middle game, and those are different again from those in the end game. But the principle remains the same:

- Look for problematic areas, or other objectives.
- Try to solve them to various degrees. One move may solve the problem completely, while others may solve it partially.
- Make now a combination and see which move does the most work and gets one closest to the final goal.

For the last evaluation one should consider one move as a move by oneself followed by one of the opponent.

One does not go over the board to consider all possible moves. This would be too exhaustive. It is like trying all different orders of elimination of n variables, each of which can be treated with a number of equations. When n is 12, this is big already, but at the next order in perturbation theory we have already that $n = 18$.

The strength of a go player lies in his/her capabilities to define projects, to solve them and to combine them. The strength of the physicist lies in his/her capability to see the combinations of equations that can be helpful.

Similarities

Now we can have a look at the similarities in the way of thinking.

- In both cases we have a well defined goal. Most points versus a program that will do our calculation.
- There are gradations in the goal. Winning by more points versus a faster program.
- At each step there are various 'moves'.
- There are rules as to what are legal moves.
- Both have an underlying search tree.
- In both cases one tries to set a reasonable objective for the next move; a project. If there is more than one project the move that solves most (weighted) objectives will be preferred.
- One can define a 'closeness' to the objective.
- Even the advanced strategic concept of 'aji' in go has its equivalent in what are the properties of the equations that are left behind for the next step.

Conclusion: if one would have a go program that could play according to project oriented reasoning, it should be possible to let its strategy unit help to derive reduction programs in perturbative field theory. And who knows what other fields of science.

The rules of Physics

So how would one do perturbative calculations with the above in mind, and assuming we have the strategy unit mentioned above?

Seeing PFT as a game, one would have to define its rules and objectives. And the initial state. The available mathematics should provide most of the rules. But considerations like how deep in ϵ one has to expand would go into it as well. The quality of the resulting code measures the size of the victory. There should be a 'language' for defining local objectives like: "I want to eliminate a given variable, but at the same time I don't want to raise another variable". Etc.

If one can do this, what would the future bring?

The work of a hypothetical future physicist:

1. set the objective: determine what physics should be done (like what calculation).
2. rules part 1: determine what mathematics might be involved. If this does not exist yet, create it or have a mathematician create it.
3. rules part 2: determine the physics boundary conditions. This is related to the objectives of course.
4. determine the initial state. Derive the equations from which to start.
5. maybe we need a measure for determining what the distance to the objective is.
6. organize the computer facilities.

Of course, in the absence of concrete results, this is pure speculation. But I do not see this as impossible! We will need more and more AI in our research if we want to have more and more terms in the perturbation expansions.