Ulrik's estimates:
https://docs.google.com/document/d/1s51XMQsox3uowVPGkIxQUqCG4VvcPieH-tmKmbhDj88/edit

[note to reviewers]: currently absent from all this is an explanation of 1) exactly what the calculators do and 2) how to use them. 1) is because I'm writing this up against a deadline, so still coding them as I wait for feedback to come in on this. In the next post, which discusses parameter choice, I'll offer to run it for anyone who's curious to give their own inputs but isn't comfortable with Python, but if you want to try them out yourself, at the time of writing, the most recent commit is dbb9a00d56039b288bb9c56fa3833157588b6031.

For the first, if you run `python3 simple_calc.py`, it will print some statements about the probability of becoming interstellar given the params in the same file (those params are currently placeholders, so put your own in if you're curious). For the second, if you run `python3 full_calc.py`, it will create a Markov Chain object, variable name `mc`, and then trigger a debugger, from which you can play around with the output using the API described here.

Note that there's a 'runtime constants' file, which determines the maximum value of certain values that are the major factor in how long the program runs (and at what point we replace the asymptotic curves with fixed values). The runtime is (I think) O(MAX_CIVILISATIONS * MAX_PROGRESS_YEARS^2), though we also do MAX_CIVILISATIONS * MAX_PLANETS^2, if that ends up being bigger. MAX_PROGRESS_YEAR_REGRESSION_STEPS limits the number of what looks like an expensive arithmetic operation - I'm currently unsure of its effect on runtime. On my 2019 Macbook Pro, the runtime was 153 seconds with the values set to the following…
MAX_PLANETS = 20
MAX_CIVILISATIONS = 30
MAX_PROGRESS_YEARS = 1000
MAX_PROGRESS_YEAR_REGRESSION_STEPS = 50
… which feels to me like about the minimum you'd want to paint a reasonable picture (some testing needed, but my guess is MAX_PROGRESS_YEARS = 10000 might be substantially better - but would take most of a day to run unless I can figure out a way to reduce the Big O value).[/note to reviewers]

Introduction

Earlier in the sequence, I argued that to date, discussions around the non-immediate-extinction part of 'existential risk' have asked insufficiently ambitious questions about 'recovery' and qualitatively imprecise definitions of 'unrecoverable' states. I suggested a family of models that

would allow us to ask, quantitatively, a more fundamental question to longtermists: ‘what is our credence that we become interstellar from various civilisational states?’ I’ve now written two pseudo-calculators that implement the models described in the previous post, both currently useable, albeit in an MVPish/alpha state.

The simpler calculator is a script available here that implements the cyclical model described in the previous post. It’s hopefully self-explanatory if you read that section of the post. The default numeric values are all placeholders, so you can add your own credences to see the effect.

The more complex calculator implements the decay/perils-focused models, which requires further design decisions and, therefore, more understanding.

The purpose of the rest of this post is to describe those design decisions and the philosophy behind them: in particular, what further parameters does it use, how does it use them, and why? I also consider some of the ways the calculator could be made both more user-friendly and more flexible to different assumptions if it were developed further. Also, I’m not super familiar with Python, so if you are, please message me if you’d be interested in doing some code review :)

Philosophy of the calculator

Expect increasing difficulty of developing technology

A core assumption behind this model is that we should expect each civilisation to preferentially use the lowest-hanging fruit, resource-wise, and thus that - with the partial exception of reboot 1, which will for the first time have leftover technology from the previous civilisation to learn from - each civilisation will tend to take longer than the previous to reach any given technology. The overall pattern is

- A high level of technology is required to become interstellar
- Reaching that level requires some amount of time at high risk of causing a collapse of civilisation, or of entirely destroying ourselves (a ‘time of perils’)
- Each collapse gives us a chance of going extinct from natural background risk before we reboot the next technological civilisation, and this chance might increase per reboot
- Each reboot increases the amount of time we spend in the subsequent time of perils
- Therefore the risk of reboot should be considered in the same category as an extinction risk - a [longtermist risk](https://forum.effectivealtruism.org/posts/zuQeTaqrjveSiSMYo/a-proposed-hierarchy-of-longtermist-concepts), if you will (hence the Github repo name); more specifically a contraction risk

The size of this risk could be negligible or enormous, depending on the values one assumes for the various relevant parameters.[^xparams]

It's possible that inside-view considerations could outweigh this effect for near-term reboots – for example, post-apocalyptic civilisations being more cautious about creating or using advanced weaponry. One could straightforwardly add these to the calculator as an if statement, but by default, it treats each reboot with the same underlying logic.

Use year-based states to represent times of perils

This is really part of the underlying model, and I discussed it somewhat in the previous post, but it's complicated enough that I think it's worth discussing further, and explaining why I picked it over a simpler implementation of (for example) availability of specific technologies.

As described in the previous post, we divide a time of perils into 'progress years'. A progress year is defined as a level of technology that a) is represented by the probabilities {x1, x2, x3...} of transitioning to other civilisational states respectively {s1,s2, s3...} during a year spent at that level of technology, b) parametrically labels the number of actual years of constant technological development which, at some abstract ideal rate of progress and given no global setbacks, we imagine it would take to reach that level of technology, and slowed by some factor for each reboot (per the previous section, and with the caveat about reboot 1 mentioned there).

This sub-model has a few benefits:
1) data about our current era heavily tends towards annualisation. For example: we have annual estimates of GDP, both at the global and country level; we have annual estimates of background extinction risk (which could become relevant given elongation of future perils states, per previous section); also the most granular existential risk estimates we have are per year (see Michael Aird's existential risk database).

So the progress year approach makes it easy to use such data to inform our credences. Similarly, other risk estimates from that database are point estimates about the probability of some outcome by some year. Point estimates don't plug in quite so neatly, since we can't straightforwardly equate <the number of years until the given date> with <a progress year progression> - but doing so maybe isn't a bad first approximation (and we can calculate a more precise way of interpreting the estimate if we think it's important).

2) it lets us consider scenarios in which a 'minor' catastrophe sets us back, say, 50 years - a blip which would usually be literally invisible in existential risk discussions, but which, on this approach, we can now think of as a small but conceivably significant longtermist risk.

3) it's agnostic about how we determine our base rate of technological progress. I chose the default values by eyeballing historical GDP, and putting us in about the ~70th progress year from 1945 (rather than the 78th, where we would have been in 2023 with positive GDP growth every year). But if, for example, you thought that the end of the space race was contingent and that in a different world it could have progressed much further, you might consider us to be in a

much less advanced progress year today (probably leading to a more optimistic set of predictions for future times of peril).
4) It gives us some scope to parameterise differential technological progress, such as by changing the relative x-stretch for certain transitional probabilities, as described in the next section.

Use an S-curve function to represent increasing probabilities of technological expansion/contraction

To capture my intuitions about risks from various technologies rising over time, I wanted a function that 'looked S-curved' but had a distinct starting point. Somewhat arbitrarily, I've used one with the form y = 1/(1 + x^-A * *e*^-x)[^xkrempel] (you can play with an implementation of this graph here), such that

- x is the progress year (and must be >= 0)
- y is the annual probability for any value of x of transitioning to the specified state given that level of technology
- *e* is the constant
- A is a somewhat nebulous 'sharpness' parameter, which determines the steepness of the centre of the S-curve (must be > 0; values between 0 and 1 remove the slow takeoff of the S). This parameter might end up being irrelevant - I normally leave it at 2 by default - but it gives the curve a natural 0 point, and allows more flexibility to otherwise tweak the shape of the curve to (for example) fit historical data

Then, to give more us more control over this curve, I added parameters to stretch and translate it, to allow us to express and generalise opinions like

- 'from an outside view, the average annual probability per year of us going extinct* won't be more than B'
- 'it would take C[^xasymptote] years for the world to max out on stockpiles of all the technologies we can currently foresee that could make us extinct*'
- 'each time civilisation has to reboot, it will take us D times longer to develop some or all technologies that make us extinct*' (with the possible exception noted above of the first reboot)
- 'there was (now that we know nuclear weaponry wouldn't ignite the atmosphere) 0 chance of us making ourselves extinct* until E years after the start of the time of perils'

* Or transitioning to a state of survival/preindustrial/industrial/multiplanetary/interstellar

Adding these considerations to the initial form gets us the inelegant (but still S-shaped) function B / (1 + 1 / (CD^k) * (x - E)^-A * *e*^ -(1 / (CD^k) * (x - E))), such that

- *k* is the number of times civilisation has 'rebooted' - that is, has regressed to a lower technology state from a time of perils or multiplanetary state (meaning we are currently in *k*=0)
- B is a stretch along the y-axis

- C[^xasymptote] is a stretch along the x-axis
- D is a multiplier on C that we apply once per civilisational reboot (so after *k* reboots, the stretch would be CD^k).
- E is a translation along the x-axis

In effect, D makes the trajectory of our current or second time of perils a blueprint for all future ones.

One could also model the result of differential technological progress in our current time of perils by (say) running the calculator with values that decrease the steepness of some particular S-curve only when k=0 (there's no native support for this, but it could be added easily if desired).

Use an exponential decay function to represent decreasing risks from settling multiple planets

In general, I expect self-sustaining settlements to provide resilience to virtually all non-extinction threats, and to most extinction threats. AI is the main exception, but would need an extension to the model (described under limitations below) to treat it properly; for now, I don't distinguish it from the other threats.

To capture my intuitions about this class of states, the majority of transition probabilities (specifically, all the transitions to lower technology states) derive from a modified exponential decay formula, with equation y = A(1 - B)^(x - C) + D, such that

- x is the number of self-sustaining settlements,
- y is the probability for any value of x of transitioning to the specified state from that number of planets
- A is a stretch parallel to the y-axis, representing the maximum probability* (for x>=2, , since 2 planets is the minimum to definitionally qualify as being in a multiplanetary state) of the transition
- B is a value between 0 and 1 representing the the steepness of the decay (so the transition probability is multiplied by (1-B) for each additional planet)*
- C is a translation parallel to the x-axis, always set to 2 by default
- D is a translation in the y-direction representing the minimum value the risk can reduce to*

* though D is applied after A and B, so will modify those values accordingly

In this class of states we don't pay any attention to the number of reboots - I assume that, for example, the availability of fossil fuels from Earth will have stopped being a limiting factor at this level of technology.

For the probability of transitioning to an interstellar state, I use the same S-curve function as above, but such that x is the number of self-sustaining settlements.

Limitations of the calculator

I have a few concerns about the calculator, some of which point to ways I would like to see it improved if I or anyone else developed it further:

1) Model uncertainty

As an implementation of the model in the previous post, all that model's limitations apply.

2) AGI

The future trajectory of AGI development seems unique among threats. Nuclear weapons, biotechnology and other advanced weaponry seem likely to pose an ongoing threat to civilisation, albeit one that might diminish exponentially as our civilisation expands. Also they could cause our civilisation to contract multiple times in much the same way.

By contrast, AGI seems likely to quickly lead to one of three outcomes: extinction, existential security, or business-as-usual with a new powerful tool. The first two aren't ongoing probabilities from having the technology - they're something that will presumably happen very quickly or not at all once we develop it (or, if creating a friendly AGI doesn't mean the risk of an unfriendly one killing us reduces to near-0, either there is some similar 'secure AGI' that does, or we don't have any chance of a long-term future).

To work this intuition into the model, I hope to introduce a separate class of states partitioning the whole of civilisation into pre- and post- development of AI states. Once AI had been developed and not caused either extinction or existential security, the risk of extinction *from AI* during both times of perils and multiplanetary states would be much lower, and the probability of going directly from a time of perils to an 'interstellar' state (which only makes sense via gaining existential security from a benevolent AI) would be 0.

For the MVP I've omitted AI as a separate consideration.

3) Usability

The S-curve logic is fairly complex, and I wonder whether it could be simplified without losing anything of value. Perhaps more importantly, the 'calculator' currently is a Python script which users will need to run locally. In an ideal world it would have a browser-based UI, though the practicality of that might be limited by the next two concerns.

4) Runtime

Because we have to model a potentially very large number of states, depending on how much precision we go for, the current runtime of the calculator is on the order of several minutes, potentially longer. This isn't a huge problem for generating a few individual estimates, but ideally we would be able to run a Monto Carlo simulation - that is, a simulation of the results of running the calculator a large number of times with somewhat randomised parameters. With the current runtimes this would be effectively impossible.

Most of this runtime comes from the implementation of the time of perils as having potentially thousands of progress years, each year a state to which you could theoretically transition from any other year in the same era. A future version of the calculator could implement a simpler version of the time of perils for simulation purposes, or allow greater separation of the intra-perils states from the bigger picture, to allow caching.

5) Function selection

The broad shape of the functions described above seem intuitively obvious to me, but people could certainly disagree. The functions shouldn't be too hard to change if you fork the code, but these straddle the boundary between 'model' and 'parameter' in a way that makes me wonder if there shouldn't be a way of giving alternatives, perhaps from a pre-determined list, as input in the parameters file.

For example, I'm really unsure how to think about evidential updates if/when we reach certain future states. If we were to reach the 100th civilisation, having neither gone extinct nor interstellar, it seems like we should strongly update that reaching either state is very unlikely from any given civilisation (at least until the expanding sun gets involved). But should the logic we use now to estimate transitional probabilities from that state take that update into account, or should we only do the update if we actually reach it?

6) Little automated testing 😔

This was just due to time restrictions - I would love to set up some tests to make the code easier to modify.

[^xasymptote] strictly speaking the graph continues to rise infinitesimally forever, and, since this is a stretch, the parameter doesn't represent a max number of years. But hopefully, by playing with it on Desmos you'll easily get an intuition for where your value of C sets the 'rough maximum' to be.

[^xkrempel] I owe this formula to Nick Krempel.

[^xparams] I’ll go into detail on the current/default parameter choices values in the next post.