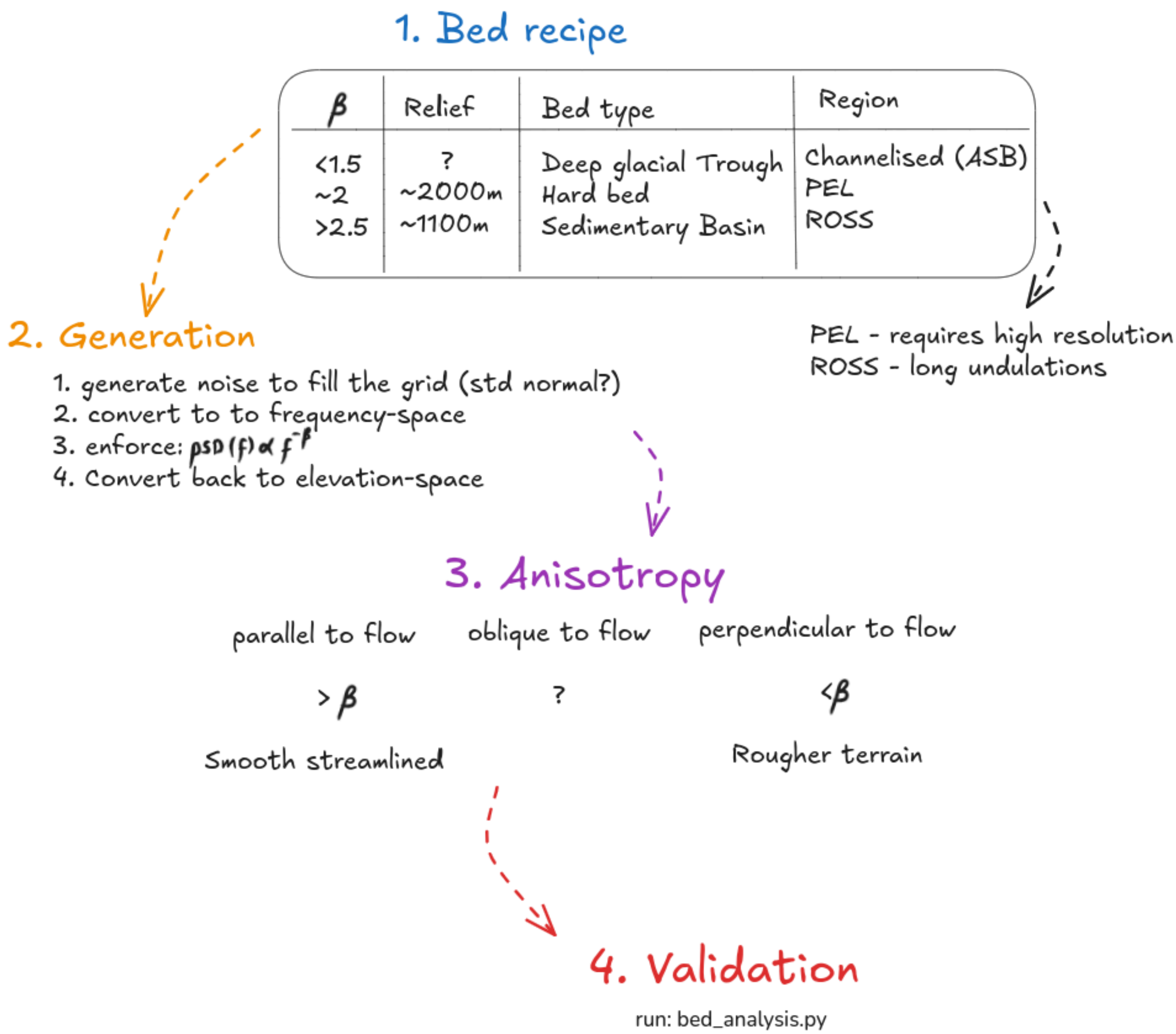
Synthetic Bedrock Generation

To generate informed artificial beds for ISSM (Ice Sheet System Model) runs, the script uses **Spectral Synthesis (Inverse FFT)** approach. This method allows the enforcement of roughness statistics and anisotropy observed in [Bedmap3 observational data analysis](). The observational analysis finds distinct statistical signatures for different geological environments. The idea behind the synthetic generator is based on "**recipes**" for each region.



2. Generation

Since the statistical analysis used the Lomb-Scargle periodogram to fit a power law, the script generates synthetic terrain by creating noise that adheres to this exact power law.

1. **Initialising the Domain:** The script defines a spatial grid with a resolution matching the observed Nyquist limit (~40–50m).

Then it creates an array of random Gaussian noise (mean 0, std 1). This represents a "flat" spectrum (**a white noise canvas**).
`white_noise = np.random.standard_normal((N, N))`

2. Applying a Fast Fourier Transform (FFT) to the noise. The script **transforms the canvas into Frequency Domain**
3. By multiplying the Fourier amplitudes by a weighting factor derived from the frequency, this **applies the spectral filter** : The Power Spectral Density (PSD) follows a power law

$$\mathrm{PSD}(f) \propto f^{-\beta}$$

The Fast Fourier Transform (FFT) works with amplitudes not power, but since $\mathrm{Power} \propto \mathrm{Amplitude}^2$, the amplitude filter is applied as the square root of the power filter.

> **TODO: Injecting Specific Bedforms (Dominant Wavelengths)**
>
> The [Bedmap3 observational data analysis] "whitened" the spectrum to find specific periodic peaks that rise above the background fractal trend.
> **Refinement Step:**
> After generating the base fractal surface (Step 2), superimpose sinusoidal waves corresponding to the "Confirmed Wavelengths" found in the text: ***Short Bedforms:*** *Add low-amplitude waves at m (typical mean wavelengths found in ASB/PEL).* **Basin-Scale Undulations:** Add larger waves at km if simulating regions like ROSS, where "candidate wavelengths" were found in this range.

4. **Inverse Transform:** Apply the Inverse FFT (IFFT) to convert the filtered spectrum back into the spatial domain (elevation). Finally, the script **normalises the resulting array** so its range (Max-Min) matches the "Vertical Relief" stats from the analysis (e.g., 1100m for soft basins like ROSS, 2143m for mountainous regions like PEL).
5. Smoothing out the result to meet slope constraints with a Gaussian filter
Surface slope drives flow; if the synthetic bed creates unrealistic surface gradients (when coupled with an ice model), the flow vectors will diverge from reality. The script implements an **adaptive physics loop** that actively modifies the bed to *force* compliance, essentially applying a "sediment drape" (smoothing out high-frequency roughness).

3. Incorporating Anisotropy (Flow Direction)

Bedrock roughness is **anisotropic**—it looks different parallel to flow vs. perpendicular to flow. This naturally generates elongated, streamlined bedforms (drumlinoids/mega-scale glacial lineations) **aligned with the flow**, and bumpy ribbing and cross-channels **perpendicular to flow

4. Validation Check

Following the logic of the Bedmap3 analysis, **verify that**:

1. The `analyse_bedrock.py` routine returns what is expected from the synthetic data.

Generated Bedrock Examples