

hacking aerosol-cloud μ -physics modeling concepts in python™

mini-workshop:
20th, 21st & 23rd Jan 2026
(12:30-3:30pm room ATG 406)

speakers:

- Sylwester Arabas @agh.edu.pl
- Emma Ware @ucdavis.edu

tools:



please bring your laptops/tablets,
browser is the only requirement

tentative agenda:

- Tue, 20th Jan:
 - SDM Monte-Carlo scheme from scratch in Python
 - lightning talk on adaptive SDM (Ware et al. 2025, arXiv:2509.05536)
- Wed, 21st Jan:
 - CCN activation model from scratch in Python (unit handling with Pint)
- Fri, 23rd Jan:
 - PySDM cloud-modeling package tutorial
 - lightning talk on immersion freezing in SDM (Arabas et al. 2025, JAMES)



photo: Eugene Timashov / National Geographic ('Cloud and ship', from AI-Petri, Crimea, Ukraine) - <https://timashovgames/>

```
1 def sdm(*,  
2     rng: np.random.Generator,  
3     xi: abc.MutableSequence[int],  
4     m: abc.MutableSequence[float],  
5     kern: abc.Callable[[float, float], float],  
6     dt: float,  
7     dv: float,  
8 ):  
9     """ SDM step assuming non-zero multiplicities """  
10    n_s = len(xi)  
11    n_pair = n_s // 2  
12    pairs = rng.permutation(n_s)[: 2 * n_pair]  
13    phi = rng.uniform(0, 1, n_pair)  
14    p_ratio = n_s * (n_s - 1) / 2 / n_pair  
15    for alpha, (j, k) in enumerate(pairs.reshape(-1, 2)):  
16        p_jk = kern(m[j], m[k]) * dt / dv  
17        if xi[j] < xi[k]:  
18            j, k = k, j  
19        p_alpha = xi[j] * p_ratio * p_jk  
20        gamma = p_alpha // 1 + (p_alpha - p_alpha // 1) > phi[alpha]  
21        if gamma != 0:  
22            gamma = min(gamma, (xi[j] / xi[k]) // 1)  
23            if xi[j] - gamma * xi[k] > 0:  
24                xi[j] -= gamma * xi[k]  
25                m[k] += gamma * m[j]  
26            else:  
27                xi[j] = xi[k] // 2  
28                xi[k] -= xi[j]  
29                m[k] += gamma * m[j]  
30                m[j] = m[k]
```