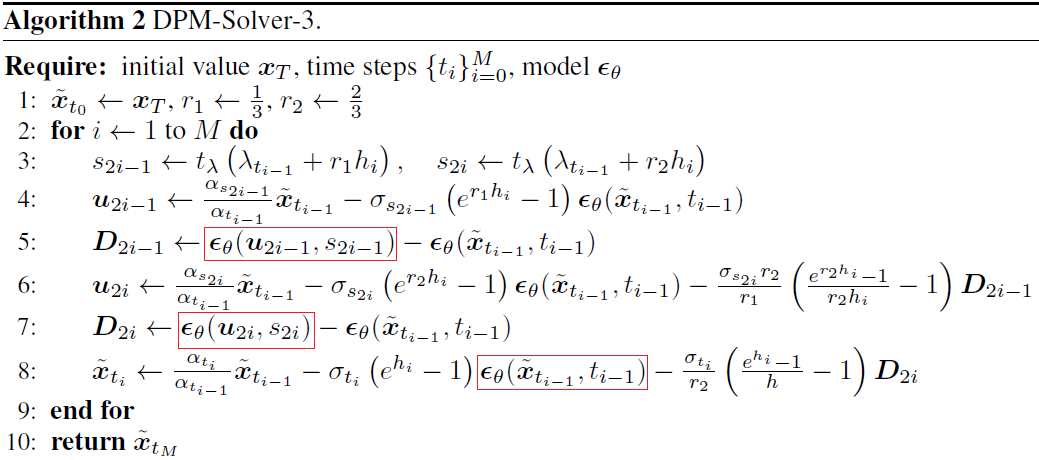
DPM Solver-3 Predefined Explanation

# Context

Paper: DPM-Solver: A Fast ODE Solver for Diffusion Probabilistic Model Sampling in Around 10 Steps. NIPS 2022.

Code: schedule/main\_schedule\_batch.py, function: calc\_loss\_order3()

Original algorithm in original paper:



# My deduction

DPM-Solver-3, in DDIM format

This is order-3 sampling. So, one sampling step involves 3 timestep. Assume the sampling step is from to , then the involved timesteps will be: .

# Conclusion

In the above formula, the variance of only depends on and .

# Calculate loss

Inner steps. From: models/dpm\_solver2\_pytorch.py, function: get\_orders\_and\_timesteps\_for\_singlestep\_solver()

|  |
| --- |
| K = steps // 3 + 1 if steps % 3 == 0:  orders = [3,] \* (K - 2) + [2, 1] elif steps % 3 == 1:  orders = [3,] \* (K - 1) + [1] else:  orders = [3,] \* (K - 1) + [2] |

## If step count % 3 == 0

Assume step count is 9. Then the inner steps are: [3, 3, 2, 1]. The sampling process will be:

|  |  |
| --- | --- |
| **x** | **Formula** |
| x9 |  |
| x8 |  |
| x7 |  |
| x6 |  |
| x5 |  |
| x4 |  |
| x3 |  |
| x2 |  |
| x1 |  |
| x0 (Generated sample) |  |

## If step count % 3 == 1

## If step count % 3 == 2