Supplementary Document: DPM-solver

Giovanni BENEDETTI DA ROSA
Cristian Alejandro CHÁVEZ BECERRA
Yann Fabio NTSAMA

Master 2 Data Science
Institut Polytechnique de Paris-École Polytechnique
Introduction to Generative Models
giovanni.benedetti-da-rosa@polytechnique.edu
cristian.chavez-becerra@polytechnique.edu

1 Introduction

In this auxiliary document accompanying the submitted report, a mathematical detailing and the results produced in the paper DPM-Solver: A Fast ODE Solver for Diffusion Probabilistic Model Sampling in Around 10 Steps will be presented. The sections containing mathematical descriptions will follow the same order as presented in the original paper.

2 Diffusion Probabilistic Models

2.1 Forward Proces and Diffusion SDEs

As described in the original paper, let $X_0 \in \mathbb{R}^D$ a random variable with unknown distribuition $q_0(x_0)$. A diffusion Probabilistic model(DPM) define a forward process $\{x_t\}_{t\in[0,T]}$ with $T > 0$ starting with x_0, such that for any $t \in [0, T]$, the distribution of x_t conditioned on x_0 satisfies:

$$q_{0t}(x_t|x_0) = \mathcal{N}(x_t|\alpha(t)x_0, \sigma^2(t)I) \tag{1}$$

where α_t, $\sigma_t \in \mathbb{R}^+$, are differentiable functions of t with bounded derivatives, and are common known as noise schedule of the DPM. Let $q_t(x_t)$ denote the marginal distribution of x_t. Diffusion Probabilistic Models (DPMs) choose noise schedules to ensure that $q_T(x_T) \approx \mathcal{N}(x_T|0, \tilde{\sigma}^2 I)$ for some $\tilde{\sigma} > 0$, and the signal-to-noise ratio (SNR) $\frac{\alpha_t^2}{\sigma_t^2}$ is strictly decreasing with respect to t.

In the paper it's stated that Kingma prove that the following stochastic differential equation(SDE) has the same distribution as stated in Eq. 1, $\forall t \in [0, T]$:

$$dx_t = f(t)x_t\, dt + g(t)\, d\mathbf{w}_t, \quad x_0 \sim q_0(x_0), \tag{2}$$

$$f(t) = \frac{d\log\alpha_t}{dt}, \quad g^2(t) = \frac{d\sigma_t^2}{dt} - 2\frac{d\log\alpha_t}{dt}\sigma_t^2. \tag{3}$$

with $\mathbf{w}_t$ represents the **standard Wiener process**.

Next, we would like to write the reverse process stated in Eq.3 from T to 0, with marginal distribution $q_T(x_T)$. To do this we check a previous reference Anderson 1982, under some conditions, where it's stated that the reverse process of this type can be generally written as:

$$dx_t = \bar{f}(x_t, t)\, dt + g(x_t, t)\, d\bar{\mathbf{w}}_t,$$

where

$$\bar{f}^i(x_t, t) = f^i(x_t, t) - \frac{1}{p(x_t, t)} \sum_{i,k} \frac{\partial}{\partial x^i}\left[p(x_t, t)g^{ik}(x_t, t)g^{ik}(x_t, t)\right].$$

In the case of the notation introduced by Anderson $p_t(x_t, t)$ represents our $q_t(x_t, t)$.

The procedure described by Anderson consider higher dimensions than our case. To be able to apply this formula we can consider that $g(t)I$. Besides that, , taking the diagonal we can rewrite this expression with the Kronecker deltas $y^{ik}(x_t, t) = y(t)\delta_{i=k}$ and $y^{jk}(x_t, t) = y(t)\delta_{j=k}$. Also,

$$\bar{f}^i(x_t,t) = f^i(x_t,t) - \frac{1}{p(x_t,t)} \sum_i \frac{\partial}{\partial x^i} \left[p(x_t,t) g^2(t)\right]$$

Next, we can say that $\bar{f}^i(x_t,t)$ is decoupled, meaning the drift can be factored into:

$$\bar{f}^i(x_t,t) = \bar{f}^i(t)(x_t),$$

and we can easily identify by definition $\nabla_x \log p(x_t,t)$.

$$\bar{f}^i(t)x_t = f^i(t)x_t - g^2(t)\nabla_x \log p(x_t,t)\big]$$

Finally we can define the reverse process, in the original notation:

$$dx_t = \left[f(t)x_t - g^2(t)\nabla_x \log q(x_t,t)\right] dt + g(t)d\bar{\mathbf{w}}_t, \quad x_t \sim q_T(x_T) \tag{4}$$

The only term that we cannot determine in Eq. 4 is the score function $\nabla_x \log q(x_t,t)$ at each time t. In practice this value is estimated ny a parametrized neural network $\epsilon_\theta(x_t,t)$ that is obtained by minimizing a objective function. By the definition of Song:

$$\mathbb{E}_t \left\{ \lambda(t) \mathbb{E}_{x(0)} \mathbb{E}_{x(t)|x(0)} \left[\left\| \mathbf{s}_\theta(\mathbf{x}(t),t) - \nabla_{\mathbf{x}(t)} \log p_{0t}(\mathbf{x}(t)|\mathbf{x}(0)) \right\|_2^2 \right] \right\} =$$

Then, we can introduce the term $\nabla_{\mathbf{x}} \log p_{0t}(\mathbf{x}(0)) = 0$.

$$\mathbb{E}_t \left\{ \lambda(t) \mathbb{E}_{x(0)} \mathbb{E}_{x(t)|x(0)} \left[\left\| \mathbf{s}_\theta(\mathbf{x}(t),t) - \nabla_{\mathbf{x}} \log p_{0t}(\mathbf{x}(0)) - \nabla_{\mathbf{x}(t)} \log p_{0t}(\mathbf{x}(t)|\mathbf{x}(0)) \right\|_2^2 \right] \right\}$$

Given that in the paper $x(0) = p_0(x)$ and $p_{0t}(x(t)|x(0))$:

$$\mathbb{E}_t \left\{ \lambda(t) \mathbb{E}_{p(0)} \mathbb{E}_{p_{0t}(x(t)|x(0))} \left[\left\| \mathbf{s}_\theta(\mathbf{x}(t),t) - \nabla_{\mathbf{x}} \log p_{0t}(\mathbf{x}(0)) - \nabla_{\mathbf{x}(t)} \log p_{0t}(\mathbf{x}(t)|\mathbf{x}(0)) \right\|_2^2 \right] \right\}$$

Next we can note that by Baye's rule we can write: $p_t(x_t) = p_0(x) p_{0t}(x(t)|x(0))$, considering i.i.d samples, we can rewrite:

$$\mathbb{E}_t \left\{ \lambda(t) \mathbb{E}_{p_t(x_t)} \left[\left\| \mathbf{s}_\theta(\mathbf{x}(t),t) - \nabla_{\mathbf{x}} \log p_t x_t \right\|_2^2 \right] \right\}$$

Finally, we apply the definition of $\mathbb{E}$ over t, considering that $\lambda(t) = -\frac{1}{2}w(t)$, and we proceed by changing $\mathbf{s}_\theta$ to $\frac{\epsilon_\theta}{\sigma_t}$. In this way, remembering the changes in the notation from p to q, we arrive at the definition of the Loss function from the DPM paper:

$$\mathcal{L}(\theta, w(t)) = \frac{1}{2} \int_0^T w(t) \mathbb{E}_{q_t(\mathbf{x}_t)} \left[\left\| \epsilon_\theta(\mathbf{x}(t),t) + \sigma_t \nabla_{\mathbf{x}} \log q_t(\mathbf{x}_t) \right\|_2^2 \right] dt. \tag{5}$$

where $\omega(t)$ is a weighting function. As $\epsilon_\theta(x_t,t)$ can also be regarded as predicting the Gaussian noise added to x_t, it is usually called the *noise prediction model*. Since the ground truth of $\epsilon_\theta(x_t,t)$ is $-\sigma_t \nabla_x \log q_t(x_t)$, DPMs replace the score function in Eq. 4 by $-\epsilon_\theta(x_t,t)/\sigma_t$ and define a parameterized reverse process (*diffusion SDE*) from time T to 0, starting with $x_T \sim \mathcal{N}(0, \tilde{\sigma}^2 I)$, as defined in the paper:

$$dx_t = \left[f(t)x_t + \frac{g^2(t)}{\sigma_t} \epsilon_\theta(x_t,t) \right] dt + g(t)d\bar{\mathbf{w}}_t, \quad x_T \sim \mathcal{N}(0, \tilde{\sigma}^2 I). \tag{6}$$

3 Customized Fast Solvers for Diffusion ODEs

As described by the authors in Section 2.2, discretizing SDEs is generally challenging in high dimensions, and achieving convergence within a few steps is difficult. On the other hand, ODEs are easier to solve, offering potential for fast sampling methods. However, a black-box ODE solver fails to converge effectively in a small number of steps. To address this, the authors propose a specialized approach to construct the solver for Diffusion ODEs, which will be thoroughly analyzed in this section.

The key insight here is that ODEs for diffusions can be simplified to an easier formulation due to its semi-linearity, which makes them easier to be approximated.

Song et al, defined the the following parameterized ODE (diffusion ODE):

$$\frac{dx_t}{dt} = h_\theta(x_t, t) := \underbrace{f(t)x_t}_{\text{Linear part}} + \underbrace{\frac{g^2(t)}{2\sigma_t}\epsilon_\theta(x_t, t)}_{\text{Nonlinear part}}, \quad x_T \sim \mathcal{N}(0, \tilde{\sigma}^2\mathbf{I}). \tag{7}$$

Let us recall the notation introduced in the previous sections. Here, x_t represents the state of the process at time t, while $h_\theta(x_t, t)$ is the drift term parameterized by θ, which governs the dynamics of the ODE. The term $f(t)$ is a time-dependent coefficient for x_t, and $g(t)$ is a time-dependent scaling factor associated with the noise term. Additionally, σ_t denotes the standard deviation parameter related to the noise. The function $\epsilon_\theta(x_t, t)$ is a learned model parameterized by θ, providing a correction term for the dynamics. Finally, the initial condition x_T is sampled from a Gaussian distribution, specifically $x_T \sim \mathcal{N}(0, \tilde{\sigma}^2\mathbf{I})$.

Giving that equation 7 is semi-linear, we can rewrite it using the *Variation of parameters*

$$\frac{dx}{dt} - f(t)x = \frac{g^2(t)}{2\sigma(t)}\epsilon_\theta(x, t)$$

Integration factor:

$$I(t) = e^{\int f(t)dt} \quad \text{thus,} \quad \dot{I}(t) = I(t)f(t)$$

Applying the integration factor:

$$I(t)\frac{dx}{dt} - I(t)f(t)x = I(t)\frac{g^2(t)}{2\sigma(t)}\epsilon_\theta(x_t, t) \therefore \frac{d}{dt}\left(I(t)x\right) = \frac{I(t)g^2(t)}{2\sigma(t)}\epsilon_\theta(x, t)$$

Integrating both sides (from s to a given t):

$$I(t)x_t = I(s)x_s + \int_s^t \frac{I(t)g^2(t)}{2\sigma(t)}\epsilon_\theta(x_\tau, \tau)\,d\tau \quad \Rightarrow \quad x_t = e^{\int_s^t f(\tau)d\tau}x_s + \int_s^t e^{\int_s^\tau f(\xi)d\xi}\frac{g^2(\tau)}{2\sigma(\tau)}\epsilon_\theta(x_\tau, \tau)\,d\tau \tag{8}$$

Next,

As observed in the original paper this idea decouples the linear part, that can be exactly computed, from the non-linear part.

The next idea is to introduce a strictly decreasing function(λ_t) defined as $\lambda_t := \log\left(\frac{\alpha_t}{\sigma_t}\right)$. This definition of strictly decreasing comes by construction once in a diffusion process.

Thus we can rewrite, $g(t)$, based on this new function:

$$g^2(t) = \frac{d\sigma_t^2}{dt} - 2\frac{dlog\alpha_t}{dt} = 2\sigma_t\frac{d\sigma_t}{dt} - 2\frac{dlog\alpha_t}{dt}$$

We can rewrite the first term as $2\sigma_t^2\frac{dlog\sigma_t}{dt}$. So, we have:

$$g^2(t) = 2\sigma_t^2\left(\frac{d\log\sigma_t}{dt} - \frac{d\log\alpha_t}{dt}\right) = -2\sigma_t^2\frac{d\lambda_t}{dt}$$

Remembering from previous section $f(t) = \frac{dlog\alpha_t}{dt}$, we can rewrite the eq uation 8.

1. Linear part:

$$e^{\int_s^t \frac{d}{d\tau}\log\alpha_\tau\,d\tau} = e^{\log\alpha_t} - e^{\log\alpha_s} = \frac{\alpha_t}{\alpha_s}.$$

2.To the Non-linear part using what we got above:

$$\int_s^t 2\sigma_\tau^2\frac{d\lambda_\tau}{d\tau}\frac{1}{\sigma_\tau}\epsilon_\theta(x_\tau, \tau)\,d\tau = \int_s^t \frac{\alpha_s}{\alpha_\tau}\frac{2\sigma(\tau)}{g^2(\tau)}\epsilon_\theta(x_\tau, \tau)\,d\tau$$

Putting both parts together:

$$x_t = \frac{\alpha_t}{\alpha_s}x_s + \int_s^t \frac{\alpha_s}{\alpha_\tau}\frac{2\sigma(\tau)}{g^2(\tau)}\epsilon_\theta(x_\tau, \tau)\,d\tau \tag{9}$$

We can perform a change of variables. Since $\lambda(t) = \lambda_t$ is a strictly decreasing function of t, it possesses an inverse function, denoted as $t_\lambda(\cdot)$, which satisfies $t = t_\lambda(\lambda(t))$.

Remembering the definition of λ_t, it is easy to see that we can rewrite it as $\frac{\sigma_t}{\alpha_t} = e^{-\lambda}$. This leads to the first proposition of the paper:

Proposition 1 Given an initial value x_s at time $s > 0$, the solution x_t at time $t \in [0, s]$ of diffusion ODEs in Equation 7:

$$x_t = \frac{\alpha_t}{\alpha_s} x_s - \alpha_t \int_{\lambda_s}^{\lambda_t} e^{-\lambda} \epsilon_\theta(x_\lambda, \lambda)\, d\lambda \tag{10}$$

The integral in 10 the exponentially weighted integral of ϵ_θ,which is very special and highly related to the exponential integrators in the literature of ODE solvers. As said in the paper, instead of using a black-box solvers for the integrals, approximating the solution at time t is equivalent to directly approximating the exponentially weighted integral of $\hat{\epsilon}_\theta$ from λ_s to λ_t.

3.1 High-Order Solvers for Diffusion ODEs

Given an initial value x_T at time T and $M + 1$ time steps $\{t_i\}_{i=0}^M$ decreasing from $t_0 = T$ to $t_M = 0$, let $\tilde{x}_{t_0} = x_T$ be the initial value. The proposed solvers use M steps to iteratively compute a sequence $\{\tilde{x}_{t_i}\}_{i=0}^M$ that approximates the true solutions at the time steps $\{t_i\}_{i=0}^M$. In particular, the final iterate $\tilde{x}_{t_M}$ approximates the true solution at time 0.

Starting with the previous value $\tilde{x}_{t_{i-1}}$ at time t_{i-1}, according to Eq. 10, the exact solution $x_{t_{i-1} \to t_i}$ at time t_i is given by:

$$x_{t_{i-1} \to t_i} = \frac{\alpha_{t_i}}{\alpha_{t_{i-1}}} \tilde{x}_{t_{i-1}} - \alpha_{t_i} \int_{\lambda_{t_{i-1}}}^{\lambda_{t_i}} e^{-\lambda} \hat{\epsilon}_\theta(\tilde{x}_\lambda, \lambda)\, d\lambda. \tag{11}$$

To compute the value $\tilde{x}_{t_i}$ for approximating $x_{t_{i-1} \to t_i}$, we need to approximate the exponentially weighted integral of $\hat{\epsilon}_\theta$ from $\lambda_{t_{i-1}}$ to λ_{t_i}. In order to do this, we denote $h_i := \lambda_{t_i} - \lambda_{t_{i-1}}$, and $\hat{\epsilon}_\theta^{(n)}(x_\lambda, \lambda) := \frac{d^n \hat{\epsilon}_\theta(x_\lambda, \lambda)}{d\lambda^n}$ as the n-th order total derivative of $\hat{\epsilon}_\theta(x_\lambda, \lambda)$ with respect to λ. We suppose that the total derivatives $\frac{d^j \hat{\epsilon}_\theta(x_\lambda, \lambda)}{d\lambda^j}$ (as a function of λ) exist and are continuous for $0 \le j \le k+1$. The function $\epsilon_\theta(x, s)$ is Lipschitz with respect to its first parameter x. For $k \ge 1$, the $(k-1)$-th order Taylor expansion of $\hat{\epsilon}_\theta(x_\lambda, \lambda)$ with respect to λ at $\lambda_{t_{i-1}}$ is:

$$\hat{\epsilon}_\theta(\hat{x}_\lambda, \lambda) = \sum_{k=0}^{n} \frac{(\lambda - \lambda_s)^k}{k!} \hat{\epsilon}_\theta^{(k)}(\hat{x}_\lambda, \lambda_s) + \mathcal{O}(h^{n+1}). \tag{12}$$

Writting it back the equation 12 in 11, considering that we can move out the sommatory by Fubini's Theorem, considering that the functions are $L1$:

$$x_{t_{i-1} \to t_i} = \frac{\alpha_{t_i}}{\alpha_{t_{i-1}}} \tilde{x}_{t_{i-1}} - \alpha_{t_i} \sum_{n=0}^{k-1} \hat{\epsilon}_\theta^{(n)}(\hat{x}_{\lambda_{t_{i-1}}}, \lambda_{t_{i-1}}) \int_{\lambda_{t_{i-1}}}^{\lambda_{t_i}} e^{-\lambda} \frac{(\lambda - \lambda_{t_{i-1}})^n}{n!}\, d\lambda + \mathcal{O}(h_i^{k+1}), \tag{13}$$

In the paper, it's stated that the integral $\int e^{-\lambda} \frac{(\lambda - \lambda_{t_{i-1}})^n}{n!}\, d\lambda$ can be derived anallitically. So, now we will show this.

We will show that the function can be written in terms of

$$\varphi_k(z) := \int_0^1 e^{(1-\delta)z} \frac{\delta^{k-1}}{(k-1)!}\, d\delta, \quad \varphi_0(z) = e^z \tag{14}$$

First, we will show that $\varphi_k(z)$, can be defined recursively.By integrating by parts:

$$\varphi_{k+1}(z) = \left[\frac{\delta^k}{k!} \cdot \frac{e^{(1-\delta)z}}{-z} \right]_0^1 + \int_0^1 e^{(1-\delta)z} \frac{\delta^{k-1}}{(k-1)!} \cdot \frac{1}{z}\, d\delta = \frac{1}{-z \cdot k!} + \frac{1}{z} \int_0^1 e^{(1-\delta)z} \frac{\delta^{k-1}}{(k-1)!}\, d\delta$$

Thus:

$$\varphi_{k+1}(z) = \frac{1}{z} \left(\varphi_k(z) - \frac{1}{k!} \right)$$

$k = 0$:

$$\varphi_k(0) = \int_0^1 \frac{\delta^{k-1}}{(k-1)!}\, d\delta = \frac{1}{k!}$$

Therefore, we finally arrive to the recursive relation, since the functions of the integration by parts are continuous differentiables:

$$\varphi_{k+1}(z) = \frac{\varphi_k(z) - \varphi_k(0)}{z}$$

Now, we are going to use it again λ_s as the lower integration limit and λ_t as the upper integration limit for simplicity. We can do a change of variables in the integral where $\lambda = \mu - \lambda_s$:

$$\int_0^{\lambda_t - \lambda_s} e^{-\lambda_s} e^{-\mu} \frac{\mu^k}{(k)!}\, d\mu$$

Then, we change the variables again $\frac{\mu}{\lambda_t - \lambda_s} = v$:

$$\int_0^1 e^{-\lambda_s} e^{-(\lambda_t - \lambda_s)v} (\lambda_t - \lambda_s)^{k+1} \frac{v^k}{k!}\, dv = h^{k+1} \int_0^1 e^{-\lambda_s} e^{-(\lambda_t - \lambda_s)v} \frac{v^k}{k!}\, dv$$

$$= \frac{h^{k+1}}{e^{\lambda_t}} \int_0^1 e^{1-v} e^{(\lambda_t - \lambda_s)} \frac{v^k}{k!}\, dv = \frac{h^{k+1}}{e^{\lambda_t}} \varphi_{k+1}(h) \tag{15}$$

Thus, we can rewrite Eq. 13 with this result, remembering $\sigma_t = \frac{\alpha_t}{e^{\lambda_t}}$:

$$x_t = \frac{\alpha_t}{\alpha_s} x_s - \sigma_t \sum_{k=0}^{n} h^{k+1} \varphi_{k+1}(h) \hat{\epsilon}_\theta^{(k)}(\hat{x}_\lambda, \lambda_s) + \mathcal{O}(h^{n+2}). \tag{16}$$

Based on that the authors described, dropping the high-order terms, we can derive a DPM-Solver-k for a specific order k. For instance, let's take $k = 1$:

$$x_{t_{i-1} \to t_i} = \frac{\alpha_{t_i}}{\alpha_{t_{i-1}}} \tilde{x}_{t_{i-1}} - \sigma_{t_i}(e^{h_i} - 1)\epsilon_\theta(\tilde{x}_{t_{i-1}}, t_{i-1}) + \mathcal{O}(h_i^2), \tag{17}$$

By dropping the high order term $\mathcal{O}(h_i^2)$.

DPM-Solver-1. Given an initial value x_T and $M + 1$ time steps $\{t_i\}_{i=0}^M$ decreasing from $t_0 = T$ to $t_M = 0$. Starting with $\tilde{x}_{t_0} = x_T$, the sequence $\{\tilde{x}_{t_i}\}_{i=1}^M$ is computed iteratively as follows:

$$\tilde{x}_{t_i} = \frac{\alpha_{t_i}}{\alpha_{t_{i-1}}} \tilde{x}_{t_{i-1}} - \sigma_{t_i} \left(e^{h_i} - 1\right) \epsilon_\theta(\tilde{x}_{t_{i-1}}, t_{i-1}),$$

where $h_i = \lambda_{t_i} - \lambda_{t_{i-1}}$.

For $k \geq 2$, approximating the first k terms of the Taylor expansion needs additional intermediate points between t and s. Given these considerations and that for solvers with $k \geq 4$ need much more intermediate points, the authors state the following theorem.

Theorem 1 (DPM-Solver-k as a k-th-order solver). *Assume $\epsilon_\theta(\mathbf{x}_t, t)$ follows the regularity conditions detailed in Appendix B.1, then for $k = 1, 2, 3$,* DPM-Solver-k *is a k-th order solver for diffusion ODEs, i.e., for the sequence $\{\tilde{\mathbf{x}}_{t_i}\}_{i=1}^M$ computed by* DPM-Solver-k*, the approximation error at time 0 satisfies*

$$\tilde{\mathbf{x}}_{t_M} - \mathbf{x}_0 = \mathcal{O}(h_{max}^k), \quad \textit{where } h_{max} = \max_{1 \leq i \leq M} (\lambda_{t_i} - \lambda_{t_{i-1}}).$$

To prove the theorem we are going to divide for 1 to 3 orders.

K = 1 First, we recall the definition of 14 and we list: $\varphi_1(h) = \frac{e^h - 1}{h}$, , $\varphi_2(h) = \frac{e^h - h - 1}{h^2}$, ,$\varphi_3(h) = \frac{e^h - h^2/2 - h - 1}{h^3}$. for K = 1, using remember the closed expression 16 and the 14 definitions to arrive at 17:

$$x_{t_{i-1} \to t_i} = \frac{\alpha_{t_i}}{\alpha_{t_{i-1}}} \tilde{x}_{t_{i-1}} - \sigma_{t_i}(e^{h_i} - 1)\epsilon_\theta(\tilde{x}_{t_{i-1}}, t_{i-1}) + \mathcal{O}(h_i^2),$$

Remembering that we assume that $\epsilon_\theta(\tilde{x}_{t_{i-1}}, t_{i-1})$ is Lipschitz with respect to x, it means that: $\epsilon_\theta(\tilde{x}_{t_{i-1}}, t_{i-1}) - \epsilon_\theta(x_{t_{i-1}}, t_{i-1})$ is of order $\mathcal{O}(\|\tilde{x}_{t_{i-1}} - x_{t_{i-1}}\|)$.:

$$\tilde{\mathbf{x}}_{t_i} = \frac{\alpha_{t_i}}{\alpha_{t_{i-1}}} \tilde{\mathbf{x}}_{t_{i-1}} - \sigma_{t_i} \left(e^{h_i} - 1\right) \left(\epsilon_\theta(\mathbf{x}_{t_{i-1}}, t_{i-1}) + \mathcal{O}(\|\tilde{\mathbf{x}}_{t_{i-1}} - \mathbf{x}_{t_{i-1}}\|)\right)$$

$$= \frac{\alpha_{t_i}}{\alpha_{t_{i-1}}}\mathbf{x}_{t_{i-1}} - \sigma_{t_i}\left(e^{h_i}-1\right)\epsilon_\theta(\mathbf{x}_{t_{i-1}}, t_{i-1}) + \mathcal{O}(\|\tilde{\mathbf{x}}_{t_{i-1}} - \mathbf{x}_{t_{i-1}}\|)$$

Considering $h_{\max} = \max_{1\leq i\leq M}(\lambda_{t_i} - \lambda_{t_{i-1}})$ to be the greatest truncation error:

$$\tilde{\mathbf{x}}_{t_i} = \mathbf{x}_{t_i} + \mathcal{O}(h_{\max}^2) + \mathcal{O}(\tilde{\mathbf{x}}_{t_{i-1}} - \mathbf{x}_{t_{i-1}}) = x_{t_0} + \mathcal{O}(h_{\max})$$

Doing this by M iterations and considering the assumption that $h_{\max} = \mathcal{O}\left(\frac{1}{M}\right)$, we can conclude the proof for $k = 1$:

$$\tilde{x}_{t_M} = x_{t_0} + \mathcal{O}(Mh_{\max}^2) = x_{t_0} + \mathcal{O}(h_{\max}).$$

K = 2 Again, we consider the following update for$0 < t < s < T$, $h := \lambda_t - \lambda_s$, and t_λ be the inverse of λ. The idea here is to repeat the same structure of the previous demonstration. Once we have proven that $\bar{x}_t = x_t + \mathcal{O}(h^3)$, we can show that $\tilde{x}_{t_i} = x_{t_i} + \mathcal{O}(h_{\max}^3) + \mathcal{O}(\tilde{x}_{t_{i-1}} - x_{t_{i-1}})$ by a similar argument. We remember then the update rule of the DPM-Solver-2.

$$s_1 = t_\lambda(\lambda_s + r_1 h), \tag{18}$$

$$\bar{\mathbf{u}} = \frac{\alpha_{s_1}}{\alpha_s}\mathbf{x}_s - \sigma_{s_1}\left(e^{r_1 h}-1\right)\epsilon_\theta(\mathbf{x}_s, s), \tag{19}$$

$$\bar{\mathbf{x}}_t = \frac{\alpha_t}{\alpha_s}\mathbf{x}_s - \sigma_t\left(e^{h}-1\right)\epsilon_\theta(\mathbf{x}_s, s) - \frac{\sigma_t}{2r_1}\left(e^{h}-1\right)\left(\epsilon_\theta(\bar{\mathbf{u}}, s_1) - \epsilon_\theta(\mathbf{x}_s, s)\right). \tag{20}$$

Taking $n = 1$ in Eq. 16, we obtain:

$$x_t = \frac{\alpha_t}{\alpha_s}x_s - \sigma_t h\varphi_1(h)\epsilon_\theta(x_s, s) - \sigma_t h^2\varphi_2(h)\hat{\epsilon}_\theta^{(1)}(\hat{x}_{\lambda_s}, \lambda_s) + \mathcal{O}(h^3) \tag{21}$$

Next, let's expand the final update equation of the DPM-2-Solver Eq.20, by Taylor expansion:

$$\bar{\mathbf{x}}_t = \frac{\alpha_t}{\alpha_s}\mathbf{x}_s - \sigma_t\left(e^{h}-1\right)\epsilon_\theta(\mathbf{x}_s, s) - \frac{\sigma_t}{2r_1}\left(e^{h}-1\right)\left(\epsilon_\theta(\bar{\mathbf{u}}, s_1) - \epsilon_\theta(\mathbf{x}_s, s)\right)$$

$$= \frac{\alpha_t}{\alpha_s}\mathbf{x}_s - \sigma_t\left(e^{h}-1\right)\epsilon_\theta(\mathbf{x}_s, s) - \frac{\sigma_t}{2r_1}\left(e^{h}-1\right)\left[\epsilon_\theta(\bar{\mathbf{u}}, s_1) - \epsilon_\theta(\mathbf{x}_{s_1}, s_1)\right]$$

$$-\frac{\sigma_t}{2r_1}\left(e^{h}-1\right)\left[\left(\lambda_{s_1} - \lambda_s\right)\hat{\epsilon}_\theta^{(1)}(\hat{\mathbf{x}}_{\lambda_s}, \lambda_s) + \mathcal{O}(h^2)\right].$$

Next, we subtract $\bar{\mathbf{x}}_t$ from $\mathbf{x}$, considering the already defined equations, remembering that $\varphi_1(h) = \frac{e^h-1}{h}$, $\varphi_2(h) = \frac{e^h-h-1}{h^2}$:

$$\bar{\mathbf{x}}_t - \mathbf{x} = \sigma_t\left[h^2\varphi_2(h) - \frac{1}{2r_1}\left(e^{h}-1\right)\left[\left(\lambda_{s_1} - \lambda_s\right)\hat{\epsilon}_\theta^{(1)}(\hat{\mathbf{x}}_{\lambda_s}, \lambda_s) + \epsilon_\theta(\bar{\mathbf{u}}, s_1) - \epsilon_\theta(\mathbf{x}_{s_1}, s_1) + \mathcal{O}(h^2)\right]\right]$$

Now, by the y the Lipschitzness of ϵ_θ, we can say, by a similar argument in the case of $k = 1$ $\|\epsilon_\theta(\bar{\mathbf{u}}, s_1) - \epsilon_\theta(\mathbf{x}_{s_1}, s_1)\| = \mathcal{O}(\|\bar{\mathbf{u}} - \mathbf{x}_{s_1}\|) = \mathcal{O}(h^2)$. Also, remembering that $e^h - 1 = \mathcal{O}(h)$, we can see that the last two terms are reduced to $\mathcal{O}(h^3)$:

$$\bar{\mathbf{x}}_t - \mathbf{x} = \sigma_t\left[h^2\varphi_2(h) - \frac{1}{2r_1}\left(e^{h}-1\right)\left(\lambda_{s_1} - \lambda_s\right)\right] + \mathcal{O}(h^3)$$

Then, noticing that:

$$h^2\varphi_2(h) - (e^h - 1)\frac{\lambda_{s_1} - \lambda_s}{2r_1} = \frac{(2e^h - h - 2 - he^h)}{2} = \mathcal{O}(h^3)$$

And, this concludes the proof.

K = 3

As in the previous proof it suffices to show that the following update has error $\bar{x}_t = x_t + \mathcal{O}(h^4)$ for $0 < t < s < T$ and $h = \lambda_s - \lambda_t$:

$$s_1 = t_\lambda(\lambda_s + r_1 h), \quad s_2 = t_\lambda(\lambda_s + r_2 h), \tag{22}$$

$$\bar{\mathbf{u}}_1 = \frac{\alpha_{s_1}}{\alpha_s}\mathbf{x}_s - \sigma_{s_1}\left(e^{r_1 h} - 1\right)\epsilon_\theta(\mathbf{x}_s, s), \tag{23}$$

$$\mathbf{D}_1 = \epsilon_\theta(\bar{\mathbf{u}}_1, s_1) - \epsilon_\theta(\mathbf{x}_s, s), \tag{24}$$

$$\bar{\mathbf{u}}_2 = \frac{\alpha_{s_2}}{\alpha_s}\mathbf{x}_s - \sigma_{s_2}\left(e^{r_2 h} - 1\right)\epsilon_\theta(\mathbf{x}_s, s) - \frac{\sigma_{s_2} r_2}{r_1}\left(\frac{e^{r_2 h} - 1}{r_2 h} - 1\right)\mathbf{D}_1, \tag{25}$$

$$\mathbf{D}_2 = \epsilon_\theta(\bar{\mathbf{u}}_2, s_2) - \epsilon_\theta(\mathbf{x}_s, s), \tag{26}$$

$$\bar{\mathbf{x}}_t = \frac{\alpha_t}{\alpha_s}\mathbf{x}_s - \sigma_t\left(e^{h} - 1\right)\epsilon_\theta(\mathbf{x}_s, s) - \frac{\sigma_t}{r_2}\left(\frac{e^{h} - 1}{h} - 1\right)\mathbf{D}_2. \tag{27}$$

First, we need to prove that $\bar{\mathbf{u}}_2 = \mathbf{x_2} + \mathcal{O}(h^3)$. We have to remember that $\frac{e^{r_2 h}-1}{r_2 h} - 1 = \mathcal{O}(h)$ and by the same arguments in the previous proofs $\bar{\mathbf{u}}_1 = \mathbf{x}_{s_1} + \mathcal{O}(h^2)$:

$$\bar{\mathbf{u}}_2 = \frac{\alpha_{s_2}}{\alpha_s}\mathbf{x}_s - \sigma_{s_2}\left(e^{r_2 h} - 1\right)\epsilon_\theta(\mathbf{x}_s, s) - \sigma_{s_2}\frac{r_2}{r_1}\left(\frac{e^{r_2 h} - 1}{r_2 h} - 1\right)(\epsilon_\theta(\mathbf{x}_{s_1}, s_1) - \epsilon_\theta(\mathbf{x}_s, s)) + \mathcal{O}(h^3)$$

$$= \frac{\alpha_{s_2}}{\alpha_s}\mathbf{x}_s - \sigma_{s_2}\left(e^{r_2 h} - 1\right)\epsilon_\theta(\mathbf{x}_s, s) - \sigma_{s_2}\frac{r_2}{r_1}\left(\frac{e^{r_2 h} - 1}{r_2 h} - 1\right)\epsilon_\theta^{(1)}(\mathbf{x}_s, s)(\lambda_{s_1} - \lambda_s) + \mathcal{O}(h^3).$$

Let $h_2 = r_2 h$, we can check that, by expanding the terms by Taylor Series:

$$\varphi_1(h_2)h_2 = e^{h_2} - 1,$$

$$\varphi_2(h_2)h_2^2 = \frac{r_2}{r_1}\left(\frac{e^{h_2} - 1}{h_2} - 1\right)(\lambda_{s_1} - \lambda_s) + \mathcal{O}(h^3),$$

Next, we rewrite the final update $\bar{\mathbf{x}}_t$:

$$\bar{x}_t = \frac{\alpha_t}{\alpha_s}x_s - \sigma_t\left(e^{h} - 1\right)\epsilon_\theta(x_s, s) - \sigma_t\frac{1}{r_2}\left(\frac{e^{h} - 1}{h} - 1\right)(\epsilon_\theta(\bar{u}_2, s_2) - \epsilon_\theta(x_s, s)).$$

Using the Lipschitzness of ϵ_θ, and the relation between $\bar{\mathbf{u}}_2$ and $\mathbf{x}_{s_2}$, Now we use that $\lambda_{s_2} - \lambda_s = r_2 h$:

$$\bar{x}_t = \frac{\alpha_t}{\alpha_s}x_s - \sigma_t\left(e^{h} - 1\right)\epsilon_\theta(x_s, s) - \sigma_t\frac{1}{r_2}\left(\frac{e^{h} - 1}{h} - 1\right)(\epsilon_\theta(x_{s_2}, s_2) - \epsilon_\theta(x_s, s)) + \mathcal{O}(h^4)$$

$$\bar{x}_t = \frac{\alpha_t}{\alpha_s}x_s - \sigma_t\left(e^{h} - 1\right)\epsilon_\theta(x_s, s) - \sigma_t\frac{1}{r_2}\left(\frac{e^{h} - 1}{h} - 1\right)\left(\epsilon_\theta^{(1)}(x_s, s)r_2 h + \frac{1}{2}\epsilon_\theta^{(2)}(x_s, s)r_2^2 h^2\right) + \mathcal{O}(h^4)$$

Taking $n = 2$ in Eq. 16, we obtain:

$$x_t = \frac{\alpha_t}{\alpha_s}x_s - \sigma_t h\varphi_1(h)\epsilon_\theta(x_s, s) - \sigma_t h^2\varphi_2(h)\epsilon_\theta^{(1)}(x_s, s) - \sigma_t h^3\varphi_3(h)\epsilon_\theta^{(2)}(x_s, s) + \mathcal{O}(h^4)$$

Finally we present the algorithms as described in the paper.

Algorithm 1 DPM-Solver-1

Require: Initial value x_T, time steps $\{t_i\}_{i=0}^{M}$, model ϵ_θ

procedure DPM-SOLVER-1$(\tilde{x}_{t_{i-1}}, t_{i-1}, t_i)$
  $h_i \leftarrow \lambda_{t_i} - \lambda_{t_{i-1}}$
  $\tilde{x}_{t_i} \leftarrow \frac{\alpha_{t_i}}{\alpha_{t_{i-1}}} \tilde{x}_{t_{i-1}} \; - \; \sigma_{t_i}\big(e^{h_i} - 1\big)\, \epsilon_\theta\big(\tilde{x}_{t_{i-1}}, t_{i-1}\big)$
  return $\tilde{x}_{t_i}$
end procedure

$\tilde{x}_{t_0} \leftarrow x_T$
for $i \leftarrow 1$ to M **do**
  $\tilde{x}_{t_i} \leftarrow$ DPM-SOLVER-1$\big(\tilde{x}_{t_{i-1}}, t_{i-1}, t_i\big)$
end for
return $\tilde{x}_{t_M}$

Fig. 1: Pseudocode for DPM-Solver-1.

Algorithm 2 DPM-Solver-2 (general)

Require: Initial value x_T, time steps $\{t_i\}_{i=0}^{M}$, model ϵ_θ, $r_1 = 0.5$

procedure DPM-SOLVER-2$(\tilde{x}_{t_{i-1}}, t_{i-1}, t_i, r_1)$
  $h_i \leftarrow \lambda_{t_i} - \lambda_{t_{i-1}}$
  $s_i \leftarrow t_\lambda\big(\lambda_{t_{i-1}} + r_1\, h_i\big)$
  $u_i \leftarrow \frac{\alpha_{s_i}}{\alpha_{t_{i-1}}}\, \tilde{x}_{t_{i-1}} \; - \; \sigma_{s_i}\big(e^{r_1\, h_i} - 1\big)\, \epsilon_\theta\big(\tilde{x}_{t_{i-1}}, t_{i-1}\big)$
  $\tilde{x}_{t_i} \leftarrow \frac{\alpha_{t_i}}{\alpha_{t_{i-1}}}\, \tilde{x}_{t_{i-1}} \; - \; \sigma_{t_i}\big(e^{h_i} - 1\big)\, \epsilon_\theta(\tilde{x}_{t_{i-1}}, t_{i-1}) \; - \; \frac{\sigma_{t_i}}{2\, r_1}\big(e^{h_i} - 1\big)\big(\epsilon_\theta(u_i, s_i) - \epsilon_\theta(\tilde{x}_{t_{i-1}}, t_{i-1})\big)$
  return $\tilde{x}_{t_i}$
end procedure

$\tilde{x}_{t_0} \leftarrow x_T$
for $i \leftarrow 1$ to M **do**
  $\tilde{x}_{t_i} \leftarrow$ DPM-SOLVER-2$\big(\tilde{x}_{t_{i-1}}, t_{i-1}, t_i, r_1\big)$
end for
return $\tilde{x}_{t_M}$

Fig. 2: Pseudocode for DPM-Solver-2.

Algorithm 3 DPM-Solver-3

Require: Initial value x_T, time steps $\{t_i\}_{i=0}^M$, model ϵ_θ, $r_1 = \frac{1}{3}$, $r_2 = \frac{2}{3}$

procedure DPM-SOLVER-3$(\tilde{x}_{t_{i-1}}, t_{i-1}, t_i, r_1, r_2)$
  $h_i \leftarrow \lambda_{t_i} - \lambda_{t_{i-1}}$
  $s_{2i-1} \leftarrow t_\lambda(\lambda_{t_{i-1}} + r_1\, h_i), \quad s_{2i} \leftarrow t_\lambda(\lambda_{t_{i-1}} + r_2\, h_i)$
  $u_{2i-1} \leftarrow \frac{\alpha_{s_{2i-1}}}{\alpha_{t_{i-1}}}\, \tilde{x}_{t_{i-1}} \;-\; \sigma_{s_{2i-1}}\big(e^{r_1\, h_i} - 1\big)\, \epsilon_\theta(\tilde{x}_{t_{i-1}}, t_{i-1})$
  $D_{2i-1} \leftarrow \epsilon_\theta(u_{2i-1}, s_{2i-1}) \;-\; \epsilon_\theta(\tilde{x}_{t_{i-1}}, t_{i-1})$
  $u_{2i} \leftarrow \frac{\alpha_{s_{2i}}}{\alpha_{t_{i-1}}}\, \tilde{x}_{t_{i-1}} \;-\; \sigma_{s_{2i}}\big(e^{r_2\, h_i} - 1\big)\epsilon_\theta(\tilde{x}_{t_{i-1}}, t_{i-1}) \;-\; \frac{\sigma_{s_{2i}}\, r_2}{r_1}\big(e^{r_2\, h_i - 1} - 1\big)\, D_{2i-1}$
  $D_{2i} \leftarrow \epsilon_\theta(u_{2i}, s_{2i}) \;-\; \epsilon_\theta(\tilde{x}_{t_{i-1}}, t_{i-1})$
  $\tilde{x}_{t_i} \leftarrow \frac{\alpha_{t_i}}{\alpha_{t_{i-1}}}\, \tilde{x}_{t_{i-1}} \;-\; \sigma_{t_i}\big(e^{h_i} - 1\big)\epsilon_\theta(\tilde{x}_{t_{i-1}}, t_{i-1}) \;-\; \frac{\sigma_{t_i}}{r_2}\big(e^{h_i - 1} - 1\big)\, D_{2i}$
  return $\tilde{x}_{t_i}$
end procedure

$\tilde{x}_{t_0} \leftarrow x_T$
for $i \leftarrow 1$ to M **do**
  $\tilde{x}_{t_i} \leftarrow$ DPM-SOLVER-3$\big(\tilde{x}_{t_{i-1}}, t_{i-1}, t_i, r_1, r_2\big)$
end for
return $\tilde{x}_{t_M}$

Fig. 3: Pseudocode for DPM-Solver-3.

References

1. J. Ho, A. Jain, and P. Abbeel, "Denoising diffusion probabilistic models," in *Advances in Neural Information Processing Systems*, vol. 33, 2020, pp. 6840–6851.
2. Y. Song, J. Sohl-Dickstein, D. P. Kingma, A. Kumar, S. Ermon, and B. Poole, "Score-based generative modeling through stochastic differential equations," in *International Conference on Learning Representations*, 2021.
3. P. Dhariwal and A. Q. Nichol, "Diffusion models beat GANs on image synthesis," in *Advances in Neural Information Processing Systems*, vol. 34, 2021, pp. 8780–8794.
4. J. Ho, T. Salimans, A. Gritsenko, W. Chan, M. Norouzi, and D. J. Fleet, "Video diffusion models," arXiv preprint arXiv:2204.03458, 2022.
5. C. Lu, Y. Zhou, F. Bao, J. Chen, C. Li, and J. Zhu, "DPM-Solver: A Fast ODE Solver for Diffusion Probabilistic Model Sampling in Around 10 Steps," *arXiv preprint arXiv:2206.00927*, 2022.
6. D. P. Kingma, T. Salimans, B. Poole, and J. Ho, "Variational diffusion models," in *Advances in Neural Information Processing Systems*, vol. 34, 2021, pp. 21696–21707.
7. B. D. O. Anderson, "Reverse-time diffusion equation models," *Stochastic Processes and Their Applications*, vol. 12, no. 3, pp. 313–326, May 1982.