1 .
v_rest，v_reset 是不是都是以细胞内为 0 电压，然后取细胞外电压？

2 .
论文的 table 2 中，有这个说明，那不是意味着 mu 随着 t 变化？我们需要这样吗？

$u_{rest} = \hat{\mu} + J_\theta \hat{r}$ [mV]	20.123	20.362	35.478	28.069	37.578	29.33	35.92	32.081
step stimulus ("thalamic input")								
RI_{ext} [mV]	0.	0.	19.	11.964	0.	0.	9.896	3.788
$\mu(t)$ [mV]	$\boxed{u_{rest} + RI_{ext} \text{ for } t \in [0.06s, 0.09s], \text{ else } u_{rest}}$							
network parameters								
N	20683	5834	21915	5479	4850	1065	14395	2948

3 . Nest 的包输入可以选 binomial 分布或 Poisson 分布，但是论文里后面在算 mesoscopic variable 的时候都用的 Poisson，所以选 binomial 是指什么呢/

4 . 预测参数的时候，一般给定哪些参数，预测哪些参数呢？

5 . 确认下，这里 tao^E_s 是指 presynapsis 是 excitatory，即针对一个 excitatory 输出而言的？（而不是当前 synapsis 是 excitatory）

modeled by a set of differential equations for a finite number of synaptic variables $y_{i,\ell}$, $\ell = 1$, . . ., L. In simulations, we model the synaptic kernel by a single exponential with constant delay $\Delta^{\alpha\beta} = \Delta$, $\epsilon^{\alpha\beta}(t) = \Theta(t - \Delta)e^{-(t-\Delta)/\tau_s^\beta}/\tau_s^\beta$, where $\Theta(t)$ denotes the Heaviside step function. The synaptic time constants are $\tau_s^E = 3$ ms and $\tau_s^I = 6$ ms for excitatory and inhibitory synapses, respectively. This kernel can be realized by a single synaptic variable $y_i^{\alpha\beta}(t)$, which obeys the first-order kinetics $\tau_s^\beta \dot{y}_i^{\alpha\beta} = -y_i^{\alpha\beta} + \sum_{j\in\Gamma^\beta} s_j^\beta(t - \Delta)$ with $\beta \in \{E, I\}$.

Generalized integrate-and-fire model. Neurons are modeled by a leaky integrate-and-

6 . 打钩的这四条代码，我有点小疑惑。J 是单个细胞的 synapsis 的 weight，而最后的 J_syn 里包含了 C0/C，相当于认为总权重（单个细胞 weight 乘上细胞连接数量）是固定的，如果 C 变化，单个细胞的 weight 就变了，意味着 J 本质是单个细胞的参考 weight，只有在 C=C0 时，J 才是真正的单个细胞的 weight。所以，让总权重（单个细胞 weight 乘上细胞连接数量）恒定的意义是什么呢？（因为要我写代码，我就直接写 J_syn = np.array([[J, -g * J], [J, -g * J]])，不会有 C 和 C0）

```
# connectivity
J = 0.3   # excitatory synaptic weight in mV if number of input connections is C0 (see below)
g = 5.    # inhibition-to-excitation ratio, -g*J is the weight for inhibitory signals
pconn = 0.2 * np.ones((M, M)) # probability of connections
delay = 1. * np.ones((M, M)) # every two populations have a delay constant
C0 = np.array([[800, 200], [800, 200]]) * 0.2   # constant reference matrix for connections
C = np.vstack((N, N)) * pconn   # numbers of input connections
J_syn = np.array([[J, -g * J], [J, -g * J]]) * C0 / C # final synaptic weights scaling as 1/C

# step current input
```

7 . 论文里求电流用的 membrane resistance， 但是代码看起来像是用的 excitatory 和 inhibitory 的 resistance（conductance），哪里弄错了吗？

$$R^{\alpha} I_{\mathrm{syn},i}^{\alpha}(t) = \tau_{\mathrm{m}}^{\alpha} \sum_{\beta=1}^{M} w^{\alpha\beta} \sum_{j \in \Gamma_i^{\beta}} \left(\epsilon^{\alpha\beta} * s_j^{\beta} \right)(t), \tag{22}$$

where R^{α} and $\tau_{\mathrm{m}}^{\alpha}$ are the membrane resistance and membrane time constant of a neuron in

```
# connect the populations
g_syn = np.ones_like(J_syn)   # synaptic conductance
g_syn[:, 0] = C_m / tau_ex
g_syn[:, 1] = C_m / tau_in
for i in range(M):
    for j in range(M):
        nest.Connect(nest_pops[j], nest_pops[i],
                     syn_spec={'weight': J_syn[i, j] * g_syn[i, j] * pconn[i, j],
                               'delay': delay[i, j]})
```

没用tao_m

8 .
我们需要算 spectrum 吗？如果需要，是不是只用基于最后的 neural activity(每个 time bin 的 firing number)直接算即可，相当于是一个以 A 为输入，spectrum 为输出的函数，这个有现成的吗？