Algorithm 1 MLE.m :Finds the minimum of $g(V,F) = -\log P(V,F|\text{data})$ by the method of preconditioned conjugate gradients. The arguments of $g(V,F)$ at the minimum give the maximum likelihood estimates for the parameters in our model. The minimization constraints the values of $F(0)$ and $F(1)$ to be 0 so that the co-variance matrix is non-singular and there is a well defined Gaussian asymptotic limit for the distribution of estimators

Require: root $\in \mathbb{R}^{B+M+1}$ or root="random", $\alpha_{\text{root}} \in \mathbb{R}^+$, counts $\in \mathcal{M}_{T\times B}(\{0,1,\cdots M\})$, gauge $\in \{0,1\}$ and $\tau \in \mathbb{R}^+$ (de-correlation time)

Ensure: $B, M, T \in \mathbb{N}^+$

$h \leftarrow \mathcal{M}_{B\times(M+1)}(\{0\})$

for $n = 0$ to M **do**

  for $b = 0$ to $B-1$ **do**

    for $t = 0$ to $T-1$ **do**

      if $(\text{counts})_{t,b}{==}n$ **then**

        $(\text{h})_{b,n}{=}(\text{h})_{b,n}{+}(\text{counts})_{t,b}/T$

      end if

    end for

  end for

end for

$N \leftarrow \{0,1,\cdots M\}$

if root=="random" **then**

  $\overrightarrow{x}_0 \leftarrow \overrightarrow{x}_{\text{rand}}$ { $\overrightarrow{x}_{\text{rand}} \in \mathbb{R}^{B+M+1}$: entries are random samples uniformly distributed in the interval [0,1] }

else

  $\overrightarrow{x}_0 \leftarrow$ root

end if

$(\overrightarrow{x}_0)_0 \leftarrow 0$ {fixing the gauge to avoid singularity in the covariance matrix}

$(\overrightarrow{x}_0)_1 \leftarrow 0$ {fixing the gauge to avoid singularity in the covariance matrix}

$\overrightarrow{\Delta}_0 \leftarrow -\overrightarrow{\nabla} g(\overrightarrow{x}_0)$ { See Algorithm 2 **logligrad.m**}

$\overrightarrow{s}_0 \leftarrow \overrightarrow{\Delta}_0$ { initial conjugate direction}

$\alpha_0 \leftarrow \arg\min_\alpha g(\overrightarrow{x}_0 + \alpha\overrightarrow{s}_0)$ { See Algorithm 3 **linemin.m**}

$\overrightarrow{x_1} = \overrightarrow{x}_0 + \alpha_0\overrightarrow{\Delta}_0$

$\epsilon \leftarrow 10^{-7}$

$m \leftarrow 0$

while $||\overrightarrow{x_n} - \overrightarrow{x_{n-1}}||/(M+B+1) \geq \epsilon$ **do**

  $m \leftarrow m+1$

  $\overrightarrow{\Delta}_n \leftarrow -\overrightarrow{\nabla} g(\overrightarrow{x}_n)$

  $\beta_n \leftarrow ||\overrightarrow{\Delta}_n||^2/||\overrightarrow{\Delta}_{n-1}||^2$ { Using the Fletcher-Reeves scaling factor}

  $\overrightarrow{s}_n \leftarrow \overrightarrow{\Delta}_n + \beta_n\overrightarrow{s}_{n-1}$

  $\alpha_n \leftarrow \arg\min_\alpha g(\overrightarrow{x}_n + \alpha\overrightarrow{s}_n)$ { See Algorithm 3 **linemin.m**}

  $\overrightarrow{x}_{n+1} \leftarrow \overrightarrow{x}_n + \alpha_n\overrightarrow{s}_n$

  $(\overrightarrow{x}_n)_0 \leftarrow 0$ {fixing the gauge to avoid singularity in the covariance matrix}

  $(\overrightarrow{x}_n)_1 \leftarrow 0$ {fixing the gauge to avoid singularity in the covariance matrix}

end while

if gauge==0 **then**

$\overrightarrow{F}^* \leftarrow (\overrightarrow{x}_m)_{0\cdots M}$ {unpacking the ML values of frustration}

$\overrightarrow{V}^* \leftarrow (\overrightarrow{x}_m)_{M+1\cdots M+B}$ {unpacking the ML values of vexation}

$\Sigma \leftarrow \mathbf{Cov}(\overrightarrow{F}^*, \overrightarrow{V}^*)$ { See Algorithm 4 **getCovMat.m**, outcome depends}

else

$\overrightarrow{F}^* \leftarrow (\overrightarrow{x}_m)_{0\cdots M} + \frac{N}{B}\sum_{j=M+1}^{M+B}((\overrightarrow{x}_m))_j$ {unpacking the ML values of frustration and performing gauge transf.}

$\overrightarrow{V}^* \leftarrow (\overrightarrow{x}_m)_{M+1\cdots M+B} - \frac{1}{B}\sum_{j=M+1}^{M+B}((\overrightarrow{x}_m))_j$ {unpacking the ML values of vexation and performing gauge transf.}

$\Sigma \leftarrow \mathbf{Cov}(\overrightarrow{F}^*, \overrightarrow{V}^*)$ { See Algorithm 4 **getCovMat.m**, outcome depends}

end if

$\sigma\overrightarrow{F}^* \leftarrow (\mathrm{diag}(\Sigma))_{0\cdots M}$ {unpacking the ML uncertainties on frustration}

$\sigma\overrightarrow{V}^* \leftarrow (\mathrm{diag}(\Sigma))_{M+1\cdots M+B}$ {unpacking the ML uncertainties on vexation}

return $\overrightarrow{F}^*, \overrightarrow{V}^*, \Sigma, \sigma\overrightarrow{F}^*, \sigma\overrightarrow{V}^*$

Algorithm 2 logligrad.m :Calculates the gradient of the likelihood function for our specific model from the analytical formula at the value of the parameters being passed as input with a give set of data

Require: $\overrightarrow{x}_n \in \mathbb{R}^{B+M+1}$, h $\in \mathcal{M}_{B\times(M+1)}(\mathbb{R})$, $N = \{0, 1, \cdots M\}$, $\tau \in \mathbb{R}^+$(decorrelation time)

Ensure: $B, M, T \in \mathbb{N}^+$

$\overrightarrow{F} \leftarrow (\overrightarrow{x}_m)_{0\cdots M}$ {unpacking the ML values of frustration}

$\overrightarrow{V} \leftarrow (\overrightarrow{x}_m)_{M+1\cdots M+B}$ {unpacking the ML values of vexation}

$(\overrightarrow{z})_b \leftarrow \left(\sum_{n=0}^{M} \exp(-(\overrightarrow{V})_b n - (\overrightarrow{F})_n)/n!\right)_{b\in\{0,\cdots,B-1\}}$

$\left(\langle\overrightarrow{N}\rangle\right)_b \leftarrow \left(\frac{1}{(\overrightarrow{z})_b}\sum_{n=0}^{M} n\exp(-(\overrightarrow{V})_b n - (\overrightarrow{F})_n)/n!\right)_{b\in\{0,\cdots,B-1\}}$

$\left(\overrightarrow{N}_{\mathrm{av}}\right)_b \leftarrow \left(\sum_{n=0}^{M} (h)_{b,n}\, n\right)_{b\in\{0,\cdots,B-1\}}$

$\left(\overrightarrow{\nabla} g\right)_{M+1\cdots M+B} \leftarrow \frac{T}{\tau} \times \left(\overrightarrow{N}_{\mathrm{av}} - \langle\overrightarrow{N}\rangle\right)$ { Where $\times$ denotes scalar multiplication}

$(P)_{b,n} = \left(\frac{1}{(\overrightarrow{z})_b}\exp(-(\overrightarrow{V})_b n - (\overrightarrow{F})_n)/n!\right)_{b\in\{0,\cdots,B-1\},n\in\{0,\cdots,M\}}$

$\left(\overrightarrow{\nabla} g\right)_{0\cdots M} \leftarrow \frac{T}{\tau} \times \left(\sum_{b=0}^{B-1}(h)_{b,n} - (P)_{b,n}\right)$ { Where $\times$ denotes scalar multiplication}

$\left(\overrightarrow{\nabla} g\right)_0 \leftarrow 0$ {fixing the gauge to avoid singularity in the covariance matrix}

$\left(\overrightarrow{\nabla} g\right)_1 \leftarrow 0$ {fixing the gauge to avoid singularity in the covariance matrix}

return $\overrightarrow{\nabla} g$

Algorithm 3 linemin.m :performs a backtracking search to determine a "sufficiently good" step size to guaratee that we are indeed minimizing the function."Sufficiently good" is determined by the Armijo-Goldstein condition

Require: $\overrightarrow{x}_n \in \mathbb{R}^{B+M+1}$, h $\in \mathcal{M}_{B\times(M+1)}(\mathbb{R})$, $N = \{0, 1, \cdots M\}$,$\tau \in \mathbb{R}^+$(de-correlation time), $\overrightarrow{s}_n \in \mathbb{R}^{B+M+1}$ (the current conjugate direction), $\overrightarrow{\nabla} g\cdot \in \mathbb{R}^{B+M+1}$(gradient at the current location),$\alpha_{\text{root}} \in \mathbb{R}^+$

Ensure: $B, M, T \in \mathbb{N}^+$

$c_1 \leftarrow 0.7$ {control parameter should be between 0 and 1}

$c_2 \leftarrow 0.0001$ {control parameter should be between 0 and 1}

$m \leftarrow \overrightarrow{\nabla} g \cdot \overrightarrow{s}_n / ||\overrightarrow{s}_n||$ {$\cdot$ denotes the usual dot product}

$t = -c_1 m$ {parameter that serves as lower bound in the Armijo-Goldstein condition}

$\alpha \leftarrow \alpha_{\text{root}}$

$\delta \leftarrow g(\overrightarrow{x}_n) - g(\overrightarrow{x}_n + \alpha\overrightarrow{s}_n/||\overrightarrow{s}_n||)$ { See Algorithm 5 **logli.m**}

while $\delta \leq \alpha t$ **do**

  $\alpha \leftarrow \alpha c_1$

  $\delta \leftarrow g(\overrightarrow{x}_n) - g(\overrightarrow{x}_n + \alpha\overrightarrow{s}_n/||\overrightarrow{s}_n||)$

end while

return α

Algorithm 4 getCovMat.m :Calculates the asymptotic Covariance Matrix as the inverse of the Fisher information matrix for our log-likelihood function

Require: $\overrightarrow{F}^* \in \mathbb{R}^{M+1}$,$\overrightarrow{V}^* \in \mathbb{R}^B$, $N = \{0, 1, \cdots M\}$,$\tau \in \mathbb{R}^+$(de-correlation time)

Ensure: $B, M, T \in \mathbb{N}^+$

$(\overrightarrow{z})_b \leftarrow \left(\sum_{n=0}^{M} \exp(-(\overrightarrow{V})_b n - (\overrightarrow{F})_n)/n!\right)_{b\in\{0,\cdots,B-1\}}$

$(P)_{b,n} = \left(\frac{1}{(\overrightarrow{z})_b} \exp(-(\overrightarrow{V})_b n - (\overrightarrow{F})_n)/n!\right)_{b\in\{0,\cdots,B-1\},n\in\{0,\cdots,M\}}$

$\left(\langle\overrightarrow{N}\rangle\right)_b \leftarrow \left(\frac{1}{(\overrightarrow{z})_b} \sum_{n=0}^{M} n \exp(-(\overrightarrow{V})_b n - (\overrightarrow{F})_n)/n!\right)_{b\in\{0,\cdots,B-1\}}$

$\left(\langle\overrightarrow{N}^2\rangle\right)_b \leftarrow \left(\frac{1}{(\overrightarrow{z})_b} \sum_{n=0}^{M} n \exp(-(\overrightarrow{V})_b n - (\overrightarrow{F})_n)/n!\right)_{b\in\{0,\cdots,B-1\}}$

$(\mathcal{H}_{FF})_{n,n'} = \frac{T}{\tau}\left(\delta_{n,n'} \sum_{n=0}^{M} (P)_{b,n} - \sum_{b=0}^{B-1} (P)_{n,b} (P)_{b,n'}\right)_{n\in\{0,\cdots,M\},n'\in\{0,\cdots,M\}}$

$(\mathcal{H}_{VV})_{b,b'} = \frac{T}{\tau}\delta_{b,b'}\left(\left(\langle\overrightarrow{N}^2\rangle\right)_b - \left(\langle\overrightarrow{N}\rangle\right)_b^2\right)_{b\in\{0,\cdots,B-1\},b'\in\{0,\cdots,B-1\}}$

$(\mathcal{H}_{VF})_{b,n} = \frac{T}{\tau}(P)_{b,n}\left(n - \left(\langle\overrightarrow{N}\rangle\right)_b\right)_{b\in\{0,\cdots,B-1\},n\in\{0,\cdots,M\}}$

if gauge==0 **then**

$(\mathcal{H})_{i\in\{0\cdots M\},j\in\{0\cdots M\}} \leftarrow \mathcal{H}_{FF}$ {constructing the Hessian by blocks}

$(\mathcal{H})_{i\in\{M+1\cdots M+B\},j\in\{M+1\cdots M+B\}} \leftarrow \mathcal{H}_{VV}$ {constructing the Hessian by blocks}

$(\mathcal{H})_{i\in\{0\cdots M\},j\in\{M+1\cdots M+B\}} \leftarrow {}^t\mathcal{H}_{VF}$ {constructing the Hessian by blocks}

$(\mathcal{H})_{i\in\{M+1\cdots M+B\},j\in\{0\cdots M\}} \leftarrow \mathcal{H}_{VF}$ {constructing the Hessian by blocks}

$\Sigma \leftarrow \mathcal{H}^+$ {+ denotes the Moore-Penrose inverse}

else

$(\mathcal{H})_{i\in\{0\cdots M\},j\in\{0\cdots M\}} \leftarrow \mathcal{H}_{FF}$ {constructing the Hessian by blocks}

$(\mathcal{H})_{i\in\{M+1\cdots M+B\},j\in\{M+1\cdots M+B\}} \leftarrow \mathcal{H}_{VV}$ {constructing the Hessian by blocks}

$(\mathcal{H})_{i\in\{0\cdots M\},j\in\{M+1\cdots M+B\}} \leftarrow {}^t\mathcal{H}_{VF}$ {constructing the Hessian by blocks}

$(\mathcal{H})_{i\in\{M+1\cdots M+B\},j\in\{0\cdots M\}} \leftarrow \mathcal{H}_{VF}$ {constructing the Hessian by blocks}

$(\mathcal{G})_{i\in\{0\cdots M\},j\in\{0\cdots M\}} \leftarrow 0$ {constructing the Gauge transformation matrix by blocks}

$(\mathcal{G})_{i\in\{M+1\cdots M+B\},j\in\{M+1\cdots M+B\}} \leftarrow -1/B$ {constructing the Gauge transformation matrix by blocks}

$(\mathcal{G})_{i\in\{0\cdots M\},j\in\{M+1\cdots M+B\}} \leftarrow i/B$ {constructing the Gauge transformation matrix by blocks}

$(\mathcal{G})_{i\in\{M+1\cdots M+B\},j\in\{0\cdots M\}} \leftarrow 0$ {constructing the Gauge transformation matrix by blocks}

$\Sigma \leftarrow \mathcal{G}\mathcal{H}^+\,{}^t\mathcal{G}$ {+ denotes the Moore-Penrose inverse}

end if

return Σ

Algorithm 5 logli.m :Calculates the value of the likelihood function for a given set of parameters and data for our specific model

Require: $\overrightarrow{x}_n \in \mathbb{R}^{B+M+1}$, h $\in \mathcal{M}_{B\times(M+1)}(\mathbb{R})$, $N = \{0, 1, \cdots M\}$,$\tau \in \mathbb{R}^+$(decorrelation time)

Ensure: $B, M, T \in \mathbb{N}^+$

$\overrightarrow{F} \leftarrow (\overrightarrow{x}_m)_{0\cdots M}$ {unpacking the ML values of frustration}

$\overrightarrow{V} \leftarrow (\overrightarrow{x}_m)_{M+1\cdots M+B}$ {unpacking the ML values of vexation}

$(\overrightarrow{z})_b \leftarrow \left(\sum_{n=0}^{M} \exp(-(\overrightarrow{V})_b n - (\overrightarrow{F})_n)/n!\right)_{b\in\{0,\cdots,B-1\}}$

$\left(\overrightarrow{F}_{\text{av}}\right)_b \leftarrow \left(\sum_{n=0}^{M} (h)_{b,n}\,(\overrightarrow{F})_n\right)_{b\in\{0,\cdots,B-1\}}$

$\left(\overrightarrow{N}_{\text{av}}\right)_b \leftarrow \left(\sum_{n=0}^{M} (h)_{b,n}\,n\right)_{b\in\{0,\cdots,B-1\}}$

$g(\overrightarrow{x}_n) \leftarrow \sum_{b=0}^{B-1} \left(\overrightarrow{F}_{\text{av}}\right)_b + \left(\overrightarrow{N}_{\text{av}}\right)_b \left(\overrightarrow{V}\right)_b + \ln(\overrightarrow{z})_b$

return $g(\overrightarrow{x}_n)$

Algorithm 6 predav.m :Predicts the average number of flies in each bin using the DFT model

Require: $\overrightarrow{F}^* \in \mathbb{R}^{M+1}$,$\overrightarrow{V}^* \in \mathbb{R}^B$, h $\in \mathcal{M}_{B\times(M+1)}(\mathbb{R})$, $N = \{0, 1, \cdots M\}$,$\hat{N}$ (total number of individuals in the system)

Ensure: $B, M, \hat{N} \in \mathbb{N}^+$

NmaxSteps $\leftarrow 130$ {number of steps before giving up on the search}

$\epsilon \leftarrow 10^{-7}$ {tolerance in the difference between the actual number of flies and the one that comes up with the guessed mu}

$c \leftarrow 0$

$\mu \leftarrow \max\left(\left\{\left(\overrightarrow{V}^*\right)_b\right\}_{b\in\{0,\cdots B-1\}}\right)$ {convenient root for the search algorithm}

$\tilde{N} \leftarrow 0$

$\delta \leftarrow \left|\max\left(\left\{\left(\overrightarrow{V}^*\right)_b\right\}_{b\in\{0,\cdots B-1\}}\right) - \min\left(\left\{\left(\overrightarrow{V}^*\right)_b\right\}_{b\in\{0,\cdots B-1\}}\right)\right|$

$a \leftarrow \mu + 2\delta$ {guess for the highest lower bound for the chemical potential}

$b \leftarrow \mu - 2\delta$ {guess for the lowest higher bound for the chemical potential}

$\mu^* \leftarrow 0$

while $c \leq$ NmaxSteps **do**

  $\mu = (a + b)/2$

  $(\overrightarrow{z})_b \leftarrow \left(\sum_{n=0}^{M} \exp(-(\overrightarrow{V})_b n - (\overrightarrow{F})_n)/n!\right)_{b\in\{0,\cdots,B-1\}}$ {normalization for probability in our model}

  $\left(\langle\overrightarrow{N}\rangle\right)_b \leftarrow \left(\frac{1}{(\overrightarrow{z})_b}\sum_{n=0}^{M} n\exp(-(\overrightarrow{V})_b n - (\overrightarrow{F})_n)/n!\right)_{b\in\{0,\cdots,B-1\}}$ {ensemble average of our model}

  $\tilde{N} \leftarrow \sum_{b=0}^{B-1}\left(\langle\overrightarrow{N}\rangle\right)_b$ {total number of flies for this iteration of μ}

  if $|\tilde{N} - \hat{N}| \leq \epsilon$ **then**

    $\mu^* \leftarrow \mu$ {real mu in case we need it}

    break

  end if

  $c \leftarrow c + 1$ {if the number of flies is not right counter goes up and we modify accordingly for the binary search}

  if $\tilde{N} < \hat{N}$ **then**

    $b \leftarrow \mu$

  else

    $a \leftarrow \mu$

  end if

end while

return $\mu, \langle\overrightarrow{N}\rangle$