

Apêndice A Especificação dos Modelos no STAN

O código com especificações dos modelos para o problema homogêneo, que foi resolvido utilizando o método analítico separação de variáveis, é dado a seguir.

```

1 stan_model_code_1 <-
2 "
3 data{
4 # Organizacao dos dados em uma lista para o Stan
5   int OT;           # Ordem de truncamento
6   int Tam_t;        # Numero de ponto no tempo
7   int nd;           # Numero de dados experimentais
8   real x_medicao;     # Posicao de medicao dos dados experimentais
9   vector[Tam_t] t;   # Vetor de tempos de simulacao
10  vector[nd] f_obs;  # Dados observados
11  vector[OT] B_i;    # Coeficiente da Solucao em Serie
12  real L;            # Comprimento da placa
13  real P_i;          # Constante \pi
14  real N_i;          # Norma
15 }
16 parameters{
17   real<lower=0> alpha;
18   real<lower=0> sigma_e;
19 }
20 model{
21   real mu_i[OT];
22   real soma[Tam_t, OT];
23   real SOL[Tam_t];
24   # Metodo Separacao de Variaveis
25   for (Nt in 1:Tam_t) {
26     for (k in 1:OT) {
27       mu_i[k] = (k*P_i);
28       soma[Nt, k] = ( B_i[k] * sin(mu_i[k] * x_medicao) * exp(-( alpha * (mu_i[k]
29         * mu_i[k]) * t[Nt])) );
30     }
31     SOL[Nt] = sum(soma[Nt, 1:OT]);
32   }
33   for(i in 1:nd) {
34     f_obs[i] ~ normal(SOL[i], sigma_e); # # Verossimilhanca normal padrao
35   }
36   alpha ~ normal(0, 10); #Priori para alpha
37   sigma_e ~ cauchy(0, 10); #Priori para sigma_e
38 }
39 "
40 model1 <- stan_model(model_code = stan_model_code_1)
41 # Valores iniciais
42 init_values <- list(
43   alpha = 5,
44   sigma_e = 0.02
45 )
46 # Fitting
47 fit_hmc1 <- sampling(model1, data = data, init = function() init_values, iter =
48   2000, warmup = 1000, chains = 1)

```

Código com especificações dos modelos para o problema não homogêneo, que foi resolvido utilizando o método de diferenças finitas, é dado a seguir.

```

1 stan_model_code_2 <-
2 "
3 # Organizacao dos dados em uma lista para o Stan
4 data{
5   int<lower=1> Nx;           # Numero de pontos no espaco
6   int<lower=1> Nt;           # Numero de pontos no tempo
7   real<lower=0> delta_t;     # Tamanho do passo no tempo
8   real<lower=0> delta_x;     # Tamanho do passo no espaco
9   real<lower=0> g;           # Fonte interna de calor
10  real<lower=0> h;           # Coeficiente de transferencia de calor
11  real<lower=0> T0;          # Temperatura inicial
12  matrix[Nx, Nt] f_obs;     # Dados observados
13 }
14 transformed data {
15   real w_min = 500000;
16 }
17 parameters {
18   real<lower=w_min> w;
19   real<lower=0.1, upper=2> k;
20   real<lower=0, upper=1> sigma_e;
21 }
22 transformed parameters {
23   real<lower=0, upper=0.5> r = (k/w) * (delta_t / (delta_x * delta_x));
24   matrix[Nx, Nt] f;
25   # Metodo de Diferencas Finitas
26   f[,1] = f_obs[,1]; # Condicao Inicial
27   for (n in 1:(Nt - 1)) {
28     for (i in 2:(Nx - 1)) {
29       f[i, n + 1] = (1 - 2 * r) * f[i, n] + r * (f[i + 1, n] + f[i - 1, n]) + (
30         delta_t / w) * g;
31     }
32     real lambda = (2 * delta_x * h / k) + 3;
33     real gamma = (2 * delta_x * h / k);
34     f[1, n + 1] = (4 * f[2, n + 1] - f[3, n + 1]) / 3; # Condicao de Contorno
35     f[Nx, n + 1] = (4 * f[Nx - 1, n + 1] - f[Nx - 2, n + 1]) / lambda +
36       (gamma * T0) / lambda; # Condicao de Contorno
37   }
38 }
39 model {
40   for (n in 2:Nt) {
41     f_obs[,n] ~ normal(f[,n], sigma_e); # Verossimilhanca normal padrao
42   }
43   w ~ normal(650000, 100000);           # Priori para w
44   k ~ normal(0.3, 0.2);                 # Priori para k
45   sigma_e ~ normal(0.1, 0.005);         # Priori para \sigma_e
46 }
47 "
48 model2 <- stan_model(model_code = stan_model_code_2)
49
50 # Valores iniciais
51 init_values <- list(
52   w = 600000,
53   k = 0.3,
54   sigma_e = 0.02
55 )
56
57 # Fitting
58 fit_hmc2 <- sampling(model2, data = data, init = function() init_values, iter =
59   2000, warmup = 1000, chains = 1)

```