NCEM PROVINCIAL MODEL CODE GUIDE

Last updated: 24 July 2020

About the NCEM Code Guide

This document is intended to serve as a guide to the National COVID-19 Epi Model (NCEM) provincial model code, supporting readers to understand its structure and logic. The model described in this document is the provincial-level NCEM model, created by the South African COVID-19 Modelling Consortium.

If there are any queries regarding the model or the code, please contact us on:
info@sacovid19mc.co.za

About the NCEM collaborators

The South African COVID-19 Modelling Consortium is a group of researchers from academic, non-profit, and government institutions across South Africa. The group is coordinated by the National Institute for Communicable Diseases, on behalf of the National Department of Health. The mandate of the group is to provide, assess and validate model projections to be used for planning purposes by the Government of South Africa. The NCEM was developed as a collaboration between the following institutions, with input from a range of experts on the Consortium:

- National Institute for Communicable Diseases
- Modelling and Simulation Hub, Africa (MASHA) at the University of Cape Town
- South African DSI-NRF Centre of Excellence in Epidemiological Modelling and Analysis (SACEMA) at Stellenbosch University
- Health Economics and Epidemiology Research Office (HE^2RO) at the University of the Witwatersrand in partnership with Boston University

Model structure

The NCEM is a stochastic compartmental transmission model to estimate the total and reported incidence of COVID-19 in South Africa at different administration levels. The model described in this document is spatially explicit at the provincial level. The model follows a generalised Susceptible - Exposed - Infectious - Recovered (SEIR) structure accounting for disease severity (asymptomatic, mild, severe, and critical cases) and different treatment pathways as shown in the figures below. Figure 1 shows the NCEM model structure. Figure 2 is an adaptation of the model diagram in Figure 1, showing how the code structure is set up around variables (**varind**), which capture the model compartments, and transitions (**traind**), which capture the model flows. Note

that the numbering of the flows in Figure 1 is set up to read from top to bottom to follow the patient pathway, whereas the numbering of the flows (**trainds**) in Figure 2 are grouped by type to follow the logic of the model code (e.g. importation & seeding, incidence, treatment seeking, natural outcomes, hospital outcomes, detection and capacity constraints).

Figure 1. NCEM model diagram

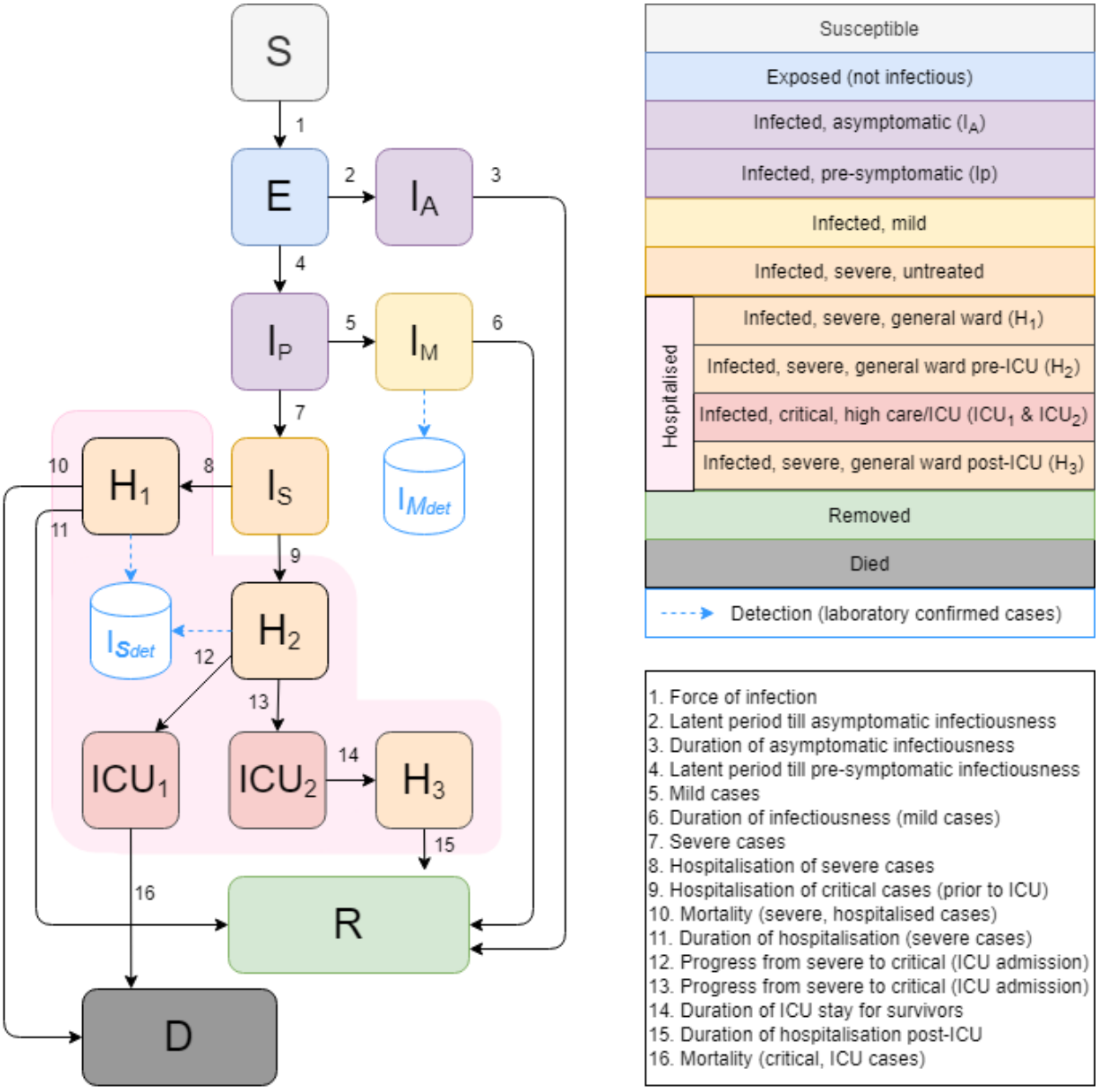
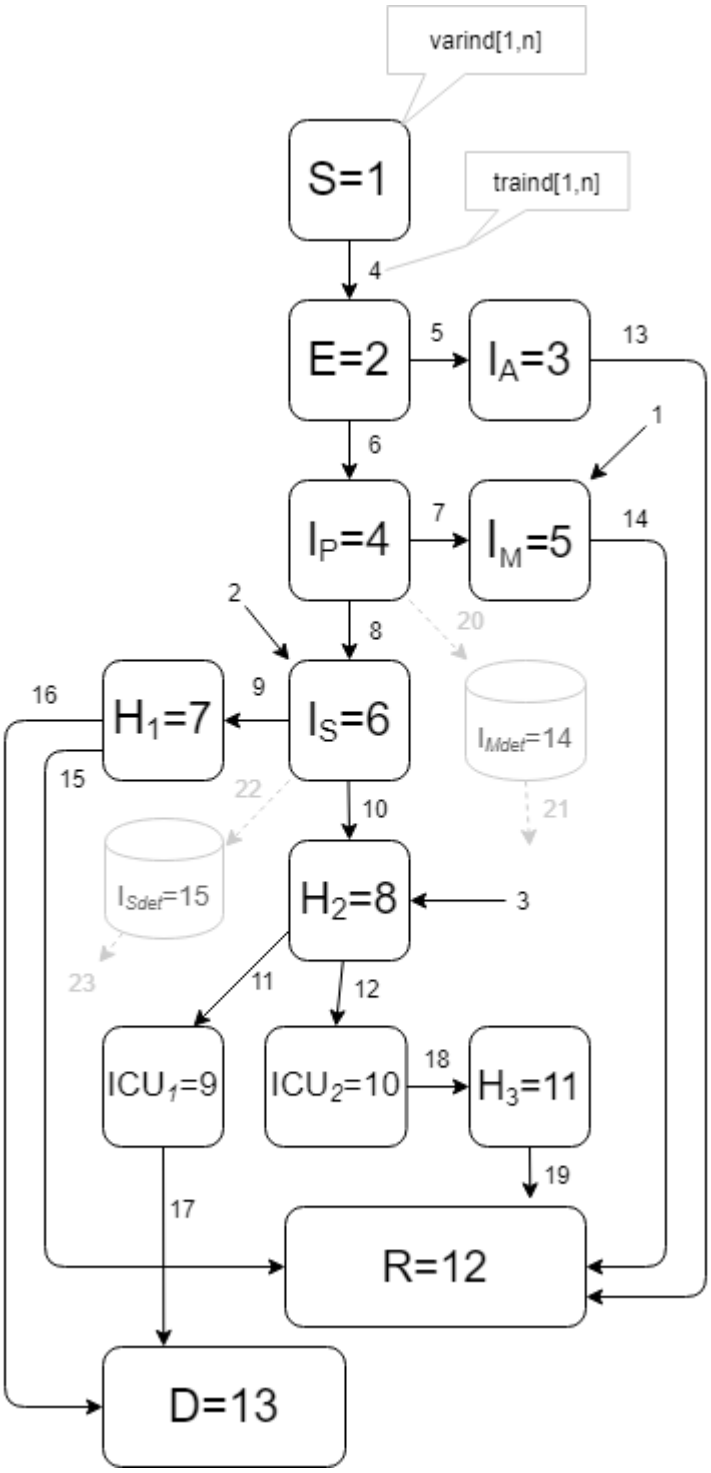


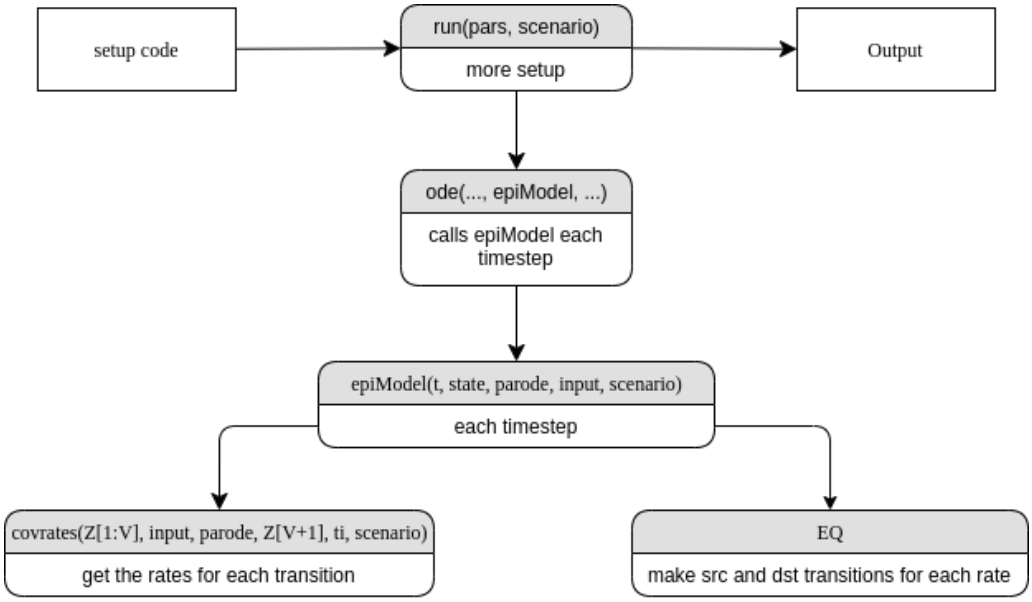
Figure 2. NCEM model diagram with **varind** and **traind** number references


varind[1,n]
S=1
traind[1,n]
4
E=2
5
I_A=3
13
1
6
I_P=4
7
I_M=5
14
2
8
20
16
H_1=7
9
I_S=6
I_{Mdef}=14
15
22
10
21
I_{Sdef}=15
H_2=8
3
23
11
12
ICU_1=9
ICU_2=10
18
H_3=11
19
17
R=12
D=13

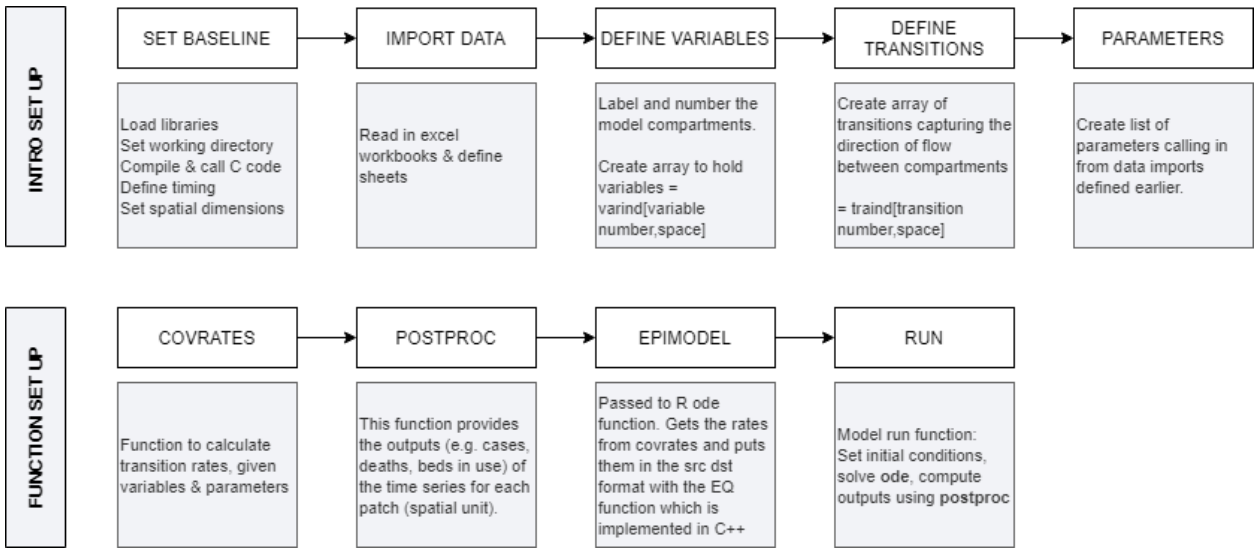
Code Structure

The overview of the code structure and set-up details are captured in Figure 3.

Figure 3. Overview of the code structure



Code set-up details:



Dimensions

The model dimensions are set based on the number of spatial units (national, provincial, districts, local municipalities), variables and transitions:

```
41 N<-9    # number of patches
42 B<-15   # number of variables per patch
43 A<-23   # number of transitions per patch
44 V<-N*B  # total number of variables
45 L<-N*A  # total number of transitions
```

This sets us up for the two main arrays: **varind** (for the number of variables) and **traind** (for the number of transitions.

In this version of the NCEM, there are 9 spatial units (representing the provinces) and no age stratification.

Compartments

Compartments are defined as variables and given a number:

```
 83 # ************************************************************************************************ #
 84 # define variables
 85 # ************************************************************************************************ #
 86 # Covid-19 Variables
 87 # 1=S: uninfected non-immune
 88 # 2=E: infected & exposed
 89 # 3=Ia: asymptomatic
 90 # 4=Ip: pre-symptomatic and infectious
 91 # 5=Im: mild and infectious
 92 # 6=Is: severe and infectious
 93 # 7=H1: severe treated in general hospital (non-ICU)
 94 # 8=H2: critical entry into general hospital (non-ICU)
 95 # 9=ICU1: critical destined to die  (ICU)
 96 # 10=ICU2: critical destined to be discharged  (ICU)
 97 # 11=H3:  critical stepdown from ICU to general hosptial (non-ICU)
 98 # 12=Removed: holds non-infectious cases, recoveries and discharges
 99 # 13=Died: holds Deaths
100 # 14=Imdet Counter: Positive confirmed mild cases | test seeking
101 # 15=Isdet Counter: Positive confirmed severe/critical cases | hospitalisation
102
```

These are stored in an array called **varind** such that the susceptible compartment S = varind[1,n]. See Figure 2.

Transitions

Transitions define the flow between compartments (see arrow numbering in Figure 2). The code separates the traditional model equations into 2 components:

1. The transitions matrix -> defines the ***direction*** of flow between compartments
2. The tranrate array (within covrates function) -> defines the ***rate*** of the flow

The model parameters and spatial connectivity are set up and passed as arguments into the covrates function.

Transition matrix:

```
# ************************************************************************************ #
# define transitions
# ************************************************************************************ #
# first transition is given without index
transitions =ssa.maketrans(V,rbind(varind[4,1], +1)) # example
for (n in 1:N){
  #Imports
  transitions[traind[1,n]]<-ssa.maketrans(V,rbind(varind[1,n], 0,varind[5,n], +1)) # imported cases -> Im=5
  transitions[traind[2,n]]<-ssa.maketrans(V,rbind(varind[1,n], 0,varind[6,n], +1)) # imported cases -> Is=6
  transitions[traind[3,n]]<-ssa.maketrans(V,rbind(varind[1,n], 0,varind[8,n], +1)) # imported cases -> H2=8
  #Incidence
  transitions[traind[4,n]]<-ssa.maketrans(V,rbind(varind[1,n], -1,varind[2,n],+1)) # incidence S to E
  transitions[traind[5,n]]<-ssa.maketrans(V,rbind(varind[2,n], -1,varind[3,n],+1)) # incubation E to Ia
  transitions[traind[6,n]]<-ssa.maketrans(V,rbind(varind[2,n], -1,varind[4,n],+1)) # incubation E to Ip
  transitions[traind[7,n]]<-ssa.maketrans(V,rbind(varind[4,n], -1,varind[5,n],+1)) # mild infection Ip to Im
  transitions[traind[8,n]]<-ssa.maketrans(V,rbind(varind[4,n], -1,varind[6,n],+1)) # severe infection Ip to Is
  #Admission to hospital
  transitions[traind[9,n]]<-ssa.maketrans(V,rbind(varind[6,n], -1,varind[7,n],+1))   # hosp severe Is to H1 (non-ICU)
  transitions[traind[10,n]]<-ssa.maketrans(V,rbind(varind[6,n], -1,varind[8,n],+1))  # hosp critical infection Is to H2 (non-ICU)
  transitions[traind[11,n]]<-ssa.maketrans(V,rbind(varind[8,n], -1,varind[9,n],+1))  # icu critical infection (destined to die) H2 to ICU1
  transitions[traind[12,n]]<-ssa.maketrans(V,rbind(varind[8,n], -1,varind[10,n],+1)) # icu critical infection (destined to be discharged) H2 to ICU2
  #Recovery
  transitions[traind[13,n]]<-ssa.maketrans(V,rbind(varind[3,n], -1,varind[12,n],+1)) # natural recovery Ia to R
  transitions[traind[14,n]]<-ssa.maketrans(V,rbind(varind[5,n], -1,varind[12,n],+1)) # natural recovery Im to R
  #Hospital outcomes
  transitions[traind[15,n]]<-ssa.maketrans(V,rbind(varind[7,n], -1,varind[12,n],+1))  # severe discharged H1 to R
  transitions[traind[16,n]]<-ssa.maketrans(V,rbind(varind[7,n], -1,varind[13,n],+1))  # severe died H1 to D
  transitions[traind[17,n]]<-ssa.maketrans(V,rbind(varind[9,n], -1,varind[13,n],+1))  # critical died ICU1 to D
  transitions[traind[18,n]]<-ssa.maketrans(V,rbind(varind[10,n], -1,varind[11,n],+1)) # stepdown to non-ICU ICU2 to H3
  transitions[traind[19,n]]<-ssa.maketrans(V,rbind(varind[11,n], -1,varind[12,n],+1)) # critical discharged H3 to R
```

"Transition 19 = (-) exit compartment 11 (H3) and (+) enter compartment 12 (R)"

Transition rate (tranrate) in covrates function:

```
256   tranrate[n,]<-c(
257     #Imports
258     pm[n]*import[n],                                  # importation to Im          1
259     ps[n]*import[n],                                  # importation to Is          2
260     (1-pm[n]-ps[n])*import[n],                        # importation to H2          3
261     #Incidence
262     foi*x[varind[1,n]],                               # incidence S to E           4
263     pa*gamma1*x[varind[2,n]],                         # incubation E to Ia         5
264     (1-pa)*gamma1*x[varind[2,n]],                     # incubation E to Ip         6
265     pm[n]*gamma2*x[varind[4,n]],                      # mild infection Ip to Im    7
266     (1-pm[n])*gamma2*x[varind[4,n]],                  # severe infection Ip to Is  8
267     #Admission to hospital
268     (1-(1-pm[n]-ps[n])/(1-pm[n]))*taus*x[varind[6,n]],  # hosp severe Is to H1       9
269     ((1-pm[n]-ps[n])/(1-pm[n]))*taus*x[varind[6,n]],    # hosp critical infection Is to H2 (non-ICU)                    10
270     pd2[n]*tauprog*x[varind[8,n]],                    # icu critical infection (destined to die) H2 to ICU1           11
271     (1-pd2[n])*tauprog*x[varind[8,n]],                # icu critical infection (destined to be discharged) H2 to ICU2 12
272     #Recovery
273     r1*x[varind[3,n]],                                # natural recovery Ia (asymp) to R   13
274     r2*x[varind[5,n]],                                # natural recovery Im to R           14
275     #Hospital outcomes
276     (1-pd1[n])*r3*x[varind[7,n]],                     # severe discharged H1 to R          15
277     pd1[n]*r3*x[varind[7,n]],                         # severe died H1 to D                16
278     mu*x[varind[9,n]],                                # critical died ICU1 to D            17
279     r4*x[varind[10,n]],                               # stepdown to non-ICU ICU2 to H3     18
280     r5*x[varind[11,n]],                               # critical discharged H3 to R        19
```

Shows flow equation to apply to traind[19]

Refers to transition number = traind[19,n]