

# The Development of Digital Twins for Cows and Dairy Farms in the Netherlands

## Abstract

With the world population rapidly increasing, having a sustainable source of nutrition is increasingly important. Dairy products may play a vital role in the global food security crisis, however greenhouse gas emissions created by the dairy industry significantly contribute to global warming and climate change which causes demand for the dairy industry to get smaller. Finding solutions that can reduce the emissions of greenhouse gasses from dairy farms plays a key role in tackling both these issues. The objective of this study was to create a digital twin model of a dairy cow and dairy herd that can be used in simulations to predict nitrogen emissions and milk yield of a dairy cow and can easily be expanded upon to include more phenotypes that can be simulated. The digital twin of the dairy cow was based on a discrete-time Markov chain model that has multiple possible states per day. Equations implemented in the model were used to calculate milk production and nitrogen emission for every step in the simulation, which were accumulated to get the total milk production and nitrogen emission over the course of the simulation. Two digital twins were simulated for 9 lactations. Cow 1, who started as a newborn, produced 95,497.5 kg milk where 90,000 kg was expected. Cow 2 started in the first day of the first lactation and produced 96,292.8 kg where 90,000 kg was expected. Further development of the digital twins may include the addition of more phenotypes, validation of the model using case studies, and improvements on the complexity of the model to allow for more accurate and faster simulations.

## Introduction

The world population has been growing quickly for the last decennia and is expected to reach 9.71 billion people in 2050 (Roser & Rodés-Guirao, 2013). Having so many people on this planet will require vast amounts of food and energy. A dilemma presenting itself here is that increasing food and energy production as it is now will increase the emission of greenhouse gasses (GHG) such as carbon dioxide (CO<sub>2</sub>), methane (CH<sub>4</sub>), and nitrous oxide (N<sub>2</sub>O). The emission of these gasses will aggravate the effects of climate change on the world by contributing to global warming. To prevent this, countries around the world are being tasked to reduce their carbon footprint, and certain climate goals have been set to prevent the effects of climate change from getting worse. The agricultural sector is being investigated as they contribute to approximately 31% of human-caused GHG emissions (Tubiello et al., 2022).

The dairy sector is crucial for the future of humanity for several reasons. Firstly, dairy products provide access to vital nutritional substances that are necessary for maintaining good health. They are a good source of proteins, essential vitamins like B2 and B12, and minerals such as calcium, which are particularly important for children and pregnant women. Secondly, the dairy sector plays a significant role in supporting rural livelihoods and economies worldwide. According to the Food and Agriculture Organization (FAO), around 600 million rural households depend on livestock, including dairy, for their livelihoods (*Transforming the Livestock Sector through the Sustainable Development Goals*, 2018). In Europe, 13% of the total production value of agricultural output is produced by dairy farming. In 2021, the EU produced 146.5 million tonnes of milk with Germany, France, the Netherlands, Italy and Poland accounting for 21.8, 16.6, 9.3, 8.9, and 8.5% respectively for a total of 65.2% of the total milk production within the EU (*Dutch Dairy in Figures 2021*, 2022). The Netherlands is a relatively large producer and exporter of dairy products in Europe for its size, responsible for 4.3% of the world dairy trade in 2021. According to statistics on the Dutch dairy sector in 2021, 6.8% of the nation's trade surplus is being produced by the dairy sector. Furthermore, 6.9% of the Dutch economy was contributed by the agricultural sector with 55.5 billion euros, of which 7.5 billion is from the dairy sector (13.5% of agricultural sector, 0.9% of total economy) (*Dutch Dairy in Figures 2021*, 2022). This demonstrates that the dairy sector is an important part of the global economy. Thirdly, the dairy sector will also play a major role in humanities efforts to provide global food security, especially in developing countries. Between 2019 and 2021, The EU exported more than 250 megatons of whole milk powder to Africa (*OECD-FAO Agricultural Outlook*, n.d.). As the world population grows and the demand for food increases, the dairy sector can provide a sustainable source of nutrition that is relatively easy to produce.

However, there is a growing concern about the environmental impact of the dairy sector. Dairy farms produce large amounts of GHG such as nitrous oxide and methane. Nitrous oxide is a potent greenhouse gas that has a global warming potential approximately 300 times greater than carbon dioxide over a 100-year time period and an atmosphere lifetime of 114 years. Moreover, it is primarily produced by agricultural activities such as use of fertilizers, manure management or burning agricultural residues. Even though nitrous oxide emissions are far lower than carbon dioxide emissions (6% compared to 79% in the US in 2021), reducing nitrous oxide emissions can significantly contribute to reaching climate goals due to its significantly higher global warming potential and long atmosphere lifetime (*Climate Change 2021*, 2021). Methane also has a greater global warming potential than carbon dioxide; approximately

25 times greater over a 100-year period and again is produced mostly by agricultural activities. Methane's shorter atmosphere lifetime of only 12 years makes it more difficult to measure emissions of it, while it also means that the majority of its environmental damage is done on short-term basis. Some nitrogen containing substances such as urea ( $(\text{NH}_2)_2\text{CO}$ ) and ammonia ( $\text{NH}_3$ ), that are mostly excreted through feces and urine and are used in fertilizers, should also be considered to be primary target substances to reduce emissions of, since these produce  $\text{CO}_2$  and nitrous oxide when reacting with water and oxygen respectively. As the effects of global warming get worse, proper control of these emissions will get more important to deal with climate change.

Finding ways to reduce nitrogen and carbon emissions from dairy farms is imperative to prevent further global warming and all its consequences. Analyses of the long-term impact of cows and dairy farms, changes in policy on farms, when to replace cows, and individual emissions of cows to find the cow with the lowest total emission, can help with this. To do this, simulations of cows can be performed.

A Markov chain Monte Carlo (MCMC) model can be used to do such simulations. A MCMC model can be defined as a model that describes a sequence of possible events, usually for a given number of steps (discrete-time Markov chain), and consists of states, stages and transition probabilities. The states are combinations of characteristics of the model, a stage or step is the time between two sequential states, and the transition probabilities are the probabilities of moving from one given state to another, which determine how the model moves through the state space. It uses random sampling with the desired distribution as its basis. It is therefore easy to replicate natural variation with it and often used in computational biology.

In the past, applications to compare output of dairy cattle have been developed. One such application is the RPO Calculator (De Vries, 2006), which used Excel spreadsheets to simulate 2 cows and compare them. This implementation however is not very maintainable and does not include a lot of the phenotypes needed to assess the effect of policies at dairy farms on global warming and climate change. Therefore, this method is not suitable for use on larger scale. Performing such simulations on a larger scale may provide new insights into the GHG emissions of dairy cattle and dairy farms, which could be used to improve efficiency and reduce GHG emissions. Many attempts have been made to create a model that can be used to simulate dairy cows, but they often failed due to the difficulty of combining the complexity of a cow with that of a simulation model.

To create a basis on which can be build, the complexity of the cow and the complexity of the simulation have to be separated. The objective of this study was to create a digital twin model for dairy cattle as a way to solve the complexity of a cow. A digital twin is a digital representation of an object that uses real-time data and can be used for simulations among other things. When creating a digital twin a few things should be kept in mind. The model should be scalable in use, easily maintainable, user friendly, and expandable in functions.

The simulation of the digital twin was done based on a discrete-time Markov chain (DTMC) model. The digital twin made in this study can be used to simulate new states, in order to predict nitrogen emissions and milk production for a dairy cow. The scope of this study was limited to cows and dairy farms in the Netherlands. The model was validated by comparing simulation results to known phenotype curves.

## Materials and Methods

### Digital Twin of Cows for Simulation

A digital twin representing a dairy cow was developed and used to simulate the dairy cows in a herd and predict the nitrogen emission of each cow over a given amount of time. The simulation of the digital twin was done based on a discrete-time Markov chain (DTMC) model that could only move in one direction as to represent the passage of time.

#### States

The states used by the digital twin included 5 characteristics, namely the life state of the cow (Open, DoNotBreed, Pregnant, Exit), the number of days in milk (DIM=0 to 1000), its lactation number (ln=0 to 9), the number of days pregnant (dp=0 to 282), and its milk production in kilograms. To reflect reality, states were created for each day. This allows for a better evaluation of measures taken, when comparing accumulated phenotypes from periods of the same length, with only small changes to parameters. This in turn may allow for tailor-made policies that minimize both nitrogen emissions and detrimental consequences for farmers.

#### State transition probabilities

The transition probabilities represented different possible daily events that a cow could have. Namely aging, dying, being replaced, getting pregnant, aborting, and calving. Below the transition probabilities are shown.

For open cows that are not yet eligible for insemination:

$$P_{S_{open,dim+1,ln,0}} = (1 - P_{mbt}) \times (1 - P_{death})$$

For open cows that are eligible for insemination and do not get pregnant:

$$P_{S_{open,dim+1,ln,0}} = (1 - P_{preg}) \times (1 - P_{mbt}) \times (1 - P_{death})$$

For open cows that are eligible for insemination and do get pregnant:

$$P_{S_{pregnant,dim+1,ln,1}} = P_{preg} \times (1 - P_{mbt}) \times (1 - P_{aic}) \times (1 - P_{death})$$

For open cows that did not get pregnant during the insemination period:

$$P_{S_{dnb,dim+1,ln,0}} = (1 - P_{preg}) \times (1 - P_{mbt}) \times P_{aic} \times (1 - P_{death})$$

For open heifers that were eligible for insemination and leaving the herd:

$$P_{S_{exit,dim+1,0,0}} = (1 - P_{preg}) \times (1 - P_{death}) + P_{death}$$

For pregnant heifers that are leaving the herd:

$$P_{S_{exit,dim+1,0,0}} = P_{abort} \times (1 - P_{death}) + P_{death}$$

For cows that are leaving the herd due to lack of production:

$$P_{S_{exit,dim+1,ln,0}} = P_{mbt}$$

For cows that are leaving the herd due to death:

$$P_{S_{exit,dim+1,ln,0}} = P_{death}$$

For pregnant cows that are not aborting pregnancy:

$$P_{S_{pregnant,dim+1,ln,dp+1}} = (1 - P_{abort}) \times (1 - P_{mbt}) \times (1 - P_{death})$$

For pregnant cows that are calving and moving to the next lactation cycle:

$$P_{S_{open,0,ln+1,0}} = P_{calv} \times (1 - P_{mbt}) \times (1 - P_{death})$$

For pregnant cows that are calving in their last lactation:

$$P_{S_{dnb,dim+1,ln+1,0}} = P_{calv} \times (1 - P_{mbt}) \times (1 - P_{death})$$

For pregnant cows that are aborting pregnancy and are eligible for re-insemination:

$$P_{S_{open,dim+1,ln,0}} = P_{abort} \times (1 - P_{mbt}) \times (1 - P_{aic}) \times (1 - P_{death})$$

For pregnant cows that are aborting pregnancy and are not eligible for re-insemination:

$$P_{S_{dnb,dim+1,ln,0}} = P_{abort} \times (1 - P_{mbt}) \times P_{aic} \times (1 - P_{death})$$

For cows that are not pregnant, not eligible for insemination and not dying:

$$P_{S_{dnb,dim+1,ln,0}} = (1 - P_{mbt}) \times (1 - P_{death})$$

Here dim = days in milk, ln = lactation number, and dp = days pregnant.  $P_{preg}$  is the daily probability of becoming pregnant, which depended on the ovulation rate, service rate and conception rate. The daily ovulation rate was set at 5.26% for heifers and 4.76% for other cows for a 19-d and 21-d estrous cycle respectively. The daily service rate was set at 85% for heifers and 65% for other cows, and the daily conception rate was set at 50% for heifers, 45% for cows in their first lactation, and 35% for other cows.  $P_{mbt}$  and  $P_{aic}$  are the probabilities of the milk production falling below the milk threshold, and the days in milk of the cow exceeding the insemination cutoff respectively. These were set to 1 whenever the respective event occurred.  $P_{calv}$  is the probability of calving, set to 1 on the last day of pregnancy.  $P_{abort}$  is the daily probability of aborting a pregnancy set to 0.0125/15 for days 30 to 45, 0.099/135 for days 46 to 180, and 0.02/(lim – 180) for the remainder of the pregnancy, where lim is the duration of a pregnancy in days for that lactation (Giordano et al., 2012). Finally,  $P_{death}$  is the daily probability of involuntary death, and was set to 0.05/365 as 5% per year. This was based on research for mortality among heifers (Zhang et al., 2019) and lactating cows (Miller et al., 2008).

## State space and replacement

The state space is the collection of all possible states that the cow can be in at any time. It can be visualised as a weighted directional graph, where the states are the nodes, the weight on the edges are the transition probabilities and the direction is the group of states that a state can transition into. Figure 1 shows examples of such a graph where in graph A, a cow does not get replaced the next day when she dies or is culled. The number inside the states represent the index in the state space. On the arrows the transition probabilities can be seen, for example state 1 has a 99.9% chance of transitioning to state 3 and state 5 has a 2.2% chance of transitioning to state 7. Graph B and C compare the difference between an implementation where a cow doesn't get replaced versus one where she does. In graph C, the cow gets replaced by a heifer the day after dying or getting culled. In this study, both implementations were created although the non-replacing implementation (A) was the main implementation. The replacement implementation (C) can be found on the reincarnation-simulation branch on the Github repository of the cow-builder package (<https://github.com/Bovi-analytics/cow-builder/tree/reincarnation-simulation>). Differences between these implementations are discussed in the discussion.

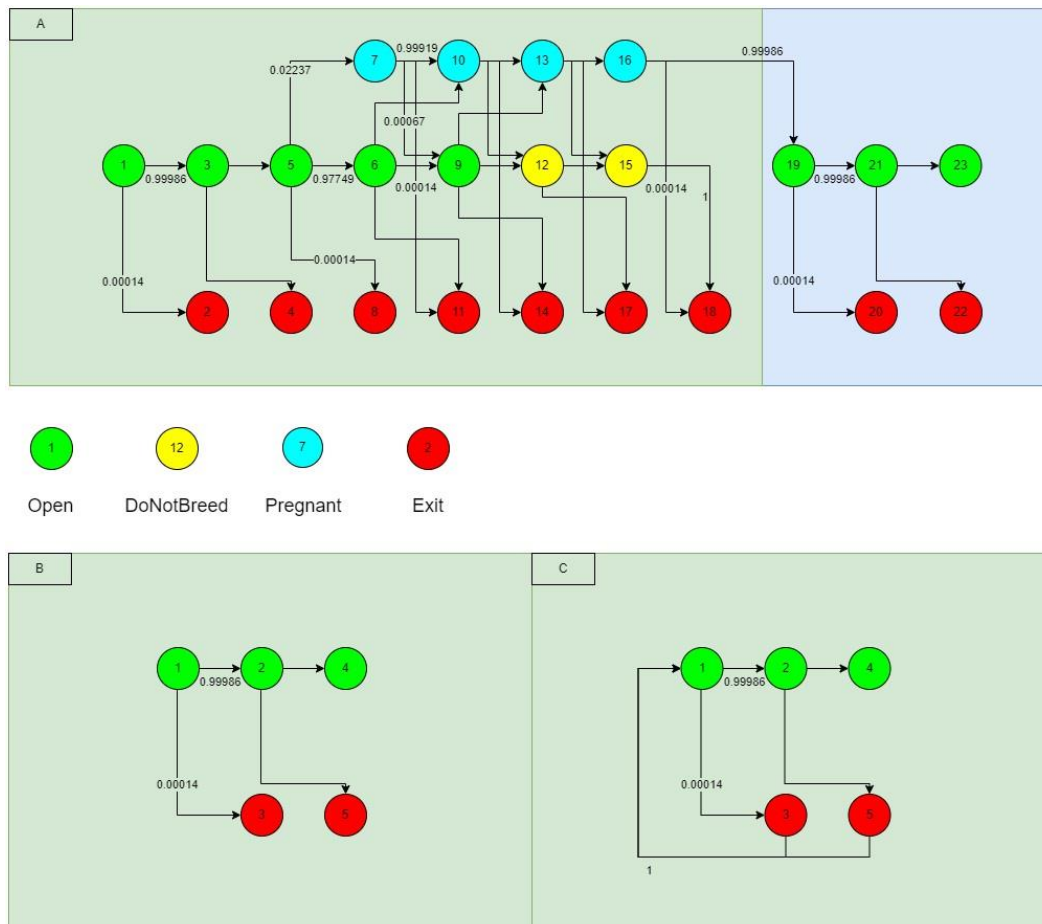


Figure 1: Visualisation a part of the state space, where graph A is the non-replacement implementation. Graph B and C show the difference between a non-replacement and replacement implementation respectively. Open and DoNotBreed states make up the middle row, Pregnant states the top row, and Exit states the bottom row. Some transition probabilities can be seen on the arrows between states. If no probability is shown it is the same as that of the last transition of the type. The number inside the state represents the index in the state space.

## The cow-builder package

The cow-builder package is the package that was created to use the digital twin of a cow and simulate it for a given time period. It was made in Python v3.10.10 and build using the package Setuptools. The cow-builder package source code can be found on Github (<https://github.com/Bovi-analytics/cow-builder/tree/v1.0>). To use the cow-builder package, two packages are required, namely chain-simulator (<https://github.com/Bovi-analytics/chain-simulator/tree/v0.3.2>) and NumPy (Harris et al., 2020). These along with other packages used for documentation generation and result visualization are described in table 5 of the appendix. Functional and non-functional requirements that were used in the process of creating the package can be found in tables 6 and 7 of the appendix respectively. By default, the cow-builder package will try to install the GPU implementation of the chain-simulator package. If the user does not have access to a dedicated NVIDIA GPU, the installation of the dependency CuPy (Okuta et al., 2017) might fail, in which case the chain-simulator package will fall back to the CPU implementation. In this study the GPU implementation was used. More information can be found in the documentation of the chain-simulator package.

Figure 2 shows a flowchart of the different steps taken when running a simulation using the cow-builder package. Steps marked in blue require the cow-builder package, while steps in orange require the chain-simulator package. To create a digital twin and simulate it, a `DigitalCow` and `DigitalHerd` object must be created, which can be imported from the `digital_cow` and `digital_herd` modules of the cow-builder package. Next, a certain number of states must be created for the cow using the `generate_total_states` function of the `DigitalCow` object.

Then, a transition matrix must be created using the chain-simulator package. If a transition matrix has already been made, it is important to note that the matrix must be made using the same parameters for creating the states of the cow. Finally, a simulation object is created and run by using the `state_vector_processor` function from the chain-simulator package. This, together with a dictionary of phenotype functions from the `digital_cow` module, are given as parameters for the `simulation_accumulator` function to get the respective phenotype values that have accumulated over the course of the simulation.

For further details on how the modules and functions in the cow-builder package work, we refer to the documentation. Sample code for how to use a module is included in the HowTo sections of each module. On the first page of the documentation, further sample code is provided of how to run an entire simulation (<https://bovi-analytics.github.io/cow-builder/>). Documentation of the chain-simulator functions used can be found on the Github Pages of the chain-simulator package (<https://bovi-analytics.github.io/chain-simulator/>).



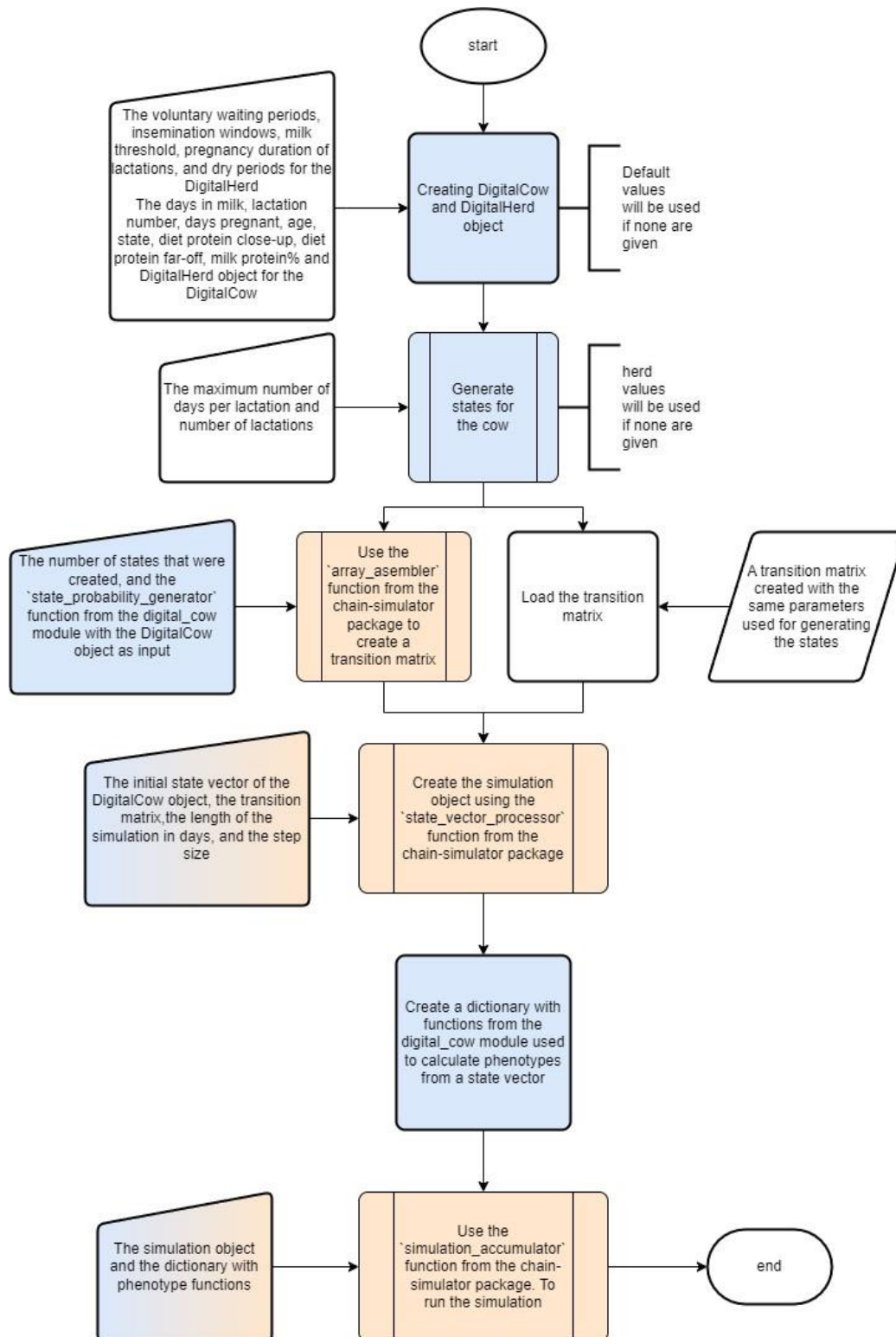


Figure 2: Flowchart of using the cow-builder and chain-simulator packages to run a simulation of a dairy cow. Steps in blue require the cow-builder package, steps in orange require the chain-simulator package, steps with a gradient require both.

## Variables & phenotypes

### Milk production

Milk production for any given day in milk dim, was calculated using the MilkBot model (Ehrlich, 2010). Table 1 shows the parameters that were used for calculating the milk production for each lactation number. For lactation number 0, the scale was set to 0 as heifers are not being milked. Other parameters were used as described in (Hostens et al., 2012).

Equation 1: MilkBot model used to fit lactation curves based on 4 parameters: scale (a), ramp (b), offset (c) and decay (d).

$$MP_{dim} = a \times \left( 1 - \frac{e^{\left[ \frac{c-dim}{b} \right]}}{2} \right) \times e^{-d \times dim}$$

Table 1: Parameters used for the MilkBot function per lactation number.

Lactation number	Scale	Ramp	Offset	Decay
0	0	1	0	1
1	41.66	29.07	0	0.001383
2	56.70	21.41	0	0.002874
>2	59.69	19.71	0	0.003262

### Body weight

For calculating the body weight of a cow two functions were used. Equation 2 was used for heifers (De Vries, 2006), where dim was the day in milk, growth rate gr had been set to 0.79, the birth weight B was set to 42 kg, and the max weight was set to 580 kg.

For lactations >0, equation 3 was used (Van Arendonk, 1985) where A was the mature live weight in kg, B was the birth weight in kg, gr was the growth rate, age was the age of the cow in days, dlw was the maximum decrease in live weight in kg, dmlw was the duration of the minimum live weight in days, p was the pregnancy parameter, and dpc was the number of days past conception - 50. If dpc was smaller than 0, it was set to 0. The parameters used for both functions (table 2), were fitted on data (Poncheki et al., 2015) to accurately represent the body weight at each day.

Equation 2: A function to determine body weight in heifers based on the day in milk (dim), a growth rate (gr), birth weight (B), and a maximum weight (maxweight)(De Vries, 2006).

$$BW_{dim} = \min_{bw} \left( \max_{bw} [B, 27.2 + \{gr \times dim\}], \maxweight \right)$$

Equation 3: A non-linear function to determine body weight in adult cows based on several parameters described in table 2.

$$BW_{dim} = A \left( 1 - \left[ 1 - \left\{ \frac{B}{A} \right\}^{1/3} \right] e^{-gr \times age} \right)^3 + \left( dlw \times \left[ \frac{dim}{dmlw} \right] \times e^{1 - \frac{dim}{dmlw}} \right) + (p^3 \times dpc^3)$$

Table 2: Parameters used in the functions described by equation 2 and 3, predicting body weight.

Lactation number	0	1	2	>2
Max weight (kg)	580			
Birth weight (kg)	42	42	42	42
Mature live weight (kg)		660	695	700
Growth rate	0.79	0.0038	0.0037	0.0037
Pregnancy parameter		0.012	0.0075	0.004
Decrease live weight (kg)		-80	-70	-60
Duration minimum live weight (d)		50	50	50

## DMI

The dry-matter intake (DMI) is the amount of feed a cow eats per day that is moisture-free. It was needed to calculate daily nitrogen emissions and calculated using equation 4. Here dim was the day in lactation for which DMI was calculated, the fat-corrected milk was the milk production of the cow on that day, and BW the body weight of the cow on that day.

Equation 4: A function that calculated dry-matter intake in kg for a given day in milk, based on the body weight (BW), fat-corrected milk (FCM), and the day in milk (dim).

$$DMI_{dim} = (0.372 \times FCM_{dim} + 0.0968 \times BW^{0.75}) \times (1 - e^{-0.192 \times [\{\frac{dim}{7}\} + 3.67]})$$

## Nitrogen emissions

Two different functions were used to calculate daily nitrogen emission of a cow. For lactating cows equation 5 was used, while for heifers equation 6 was used (Committee on Nutrient Requirements of Dairy Cattle et al., 2021). Here, N is the amount of nitrogen emitted from feces and urine in g, DMI is the dry-matter-intake in kg, DMI<sub>cp</sub> is the percentage of crude protein in the diet, MP is the milk production in kg and MP<sub>cp</sub> is the percentage of crude protein in the milk.

Equation 5: A function that calculated nitrogen emission in g, based on the dry-matter intake (DMI), dry-matter intake protein percentage (DMI<sub>cp</sub>), milk production (MP), and milk protein percentage (MP<sub>cp</sub>).

$$N = \left( \frac{DMI \times DMI_{cp}}{0.625} \right) - \left( \frac{MP \times MP_{cp}}{0.638} \right) - 5$$

Equation 6: A function that calculated nitrogen emission in g, based on the dry-matter intake (DMI), and dry-matter intake protein percentage (DMI<sub>cp</sub>)

$$N = 15.1 + \left( 0.83 \times \left[ \frac{DMI \times \frac{DMI_{cp}}{100}}{0.625} \right] \right)$$

When a heifer was in the first half of their voluntary waiting period, a lactating cow less than 100 days in lactation, or a lactating cow in the second half of their dry period,  $DMI_{cp}$  was set to 16%. For heifers that were halfway through the voluntary waiting period and lactating cows in the first half of their dry period,  $DMI_{cp}$  was set to 14%. Finally, for lactating cows that were more than 100 days in lactation but not yet in their dry period,  $DMI_{cp}$  was set to 15%.  $MP_{cp}$  was set at 3.4% for all cows.

### Cumulative phenotype

The equations described above were used to calculate the cumulative milk yield and nitrogen emission. To reduce the amount of computational time required to perform the simulation, these were put into separate cached functions so that identical calculations would take less time to perform. The functions were used by the ``vector_milk_production`` and ``vector_nitrogen_emission`` functions to calculate milk production and nitrogen emission respectively, for every non-Exit state that the digital twin could be in at a given point in time of the simulation. The averages of these outputs were taken as the milk production and nitrogen emission for that day in the simulation and added to the accumulators.

### Validation of the model

To validate the model, a simulation was performed for 2 cows, one starting as a heifer, and the other at the first day of her 1<sup>st</sup> lactation. Cow parameters for cow 1 and cow 2 are shown in table 3. Herd parameters were kept at default values. States were generated for 9 lactations with a maximum day in milk of 1000. The simulation was run for 4900 days, with a step size or interval of 14 days.

*Table 3: Parameters for the DigitalCow objects used in a simulation to validate the model.*

Parameters	Days in milk	Lactation number	Days pregnant	Diet cp close-up	Diet cp far-off	Milk cp	Age	Herd	State
Cow 1	0	0	0	160	140	3.4	0	Herd 1	Open
Cow 2	0	1	0	160	140	3.4	660	Herd 1	Open

## Results

### Validation of the model

Validation of the model was done by comparing intermediate results during the simulation, to the expected phenotype curves for each lactation cycle that was simulated.

Figure 3 shows the average milk production of 2 simulated cows during a simulation of 9 lactations where cow 1 started as a heifer and cow 2 started in her first lactation. The peaks in figure 3 can be compared to those in figure 4, which plotted the first 4 lactation curves of an average cow. Interesting to note in figure 3 is that around 2000 days old the lactation curves splits into two peaks. This is due to that at this point in the simulation the cows can be in either their 4<sup>th</sup> or 5<sup>th</sup> lactation. This overlap causes the lactations to merge when looking at average milk production which is in line with expected results.

In the last lactation of the simulation the cow has no new lactation to go into, which causes only the DoNotBreed and corresponding Exit states from after the pregnancy to remain. This is visualized in figure 3 around 4800 days for cow 1 and 4000 days for cow 2 and is also an expected result.

When comparing cow 1 to cow 2 in figure 3, cow 2 has a higher average milk production in the first few lactations. This can be attributed to the reduced variation in possible states, since cow 2 started at a later point in time. As time passes in the simulation, the variation in possible states increases and the probability of the cow being in any of these states decreases. After a certain amount of time the simulation reaches a steady state, which is when the difference in net output stays the same. This can be seen in the last lactation of both cows in the simulation, with only a shift in time due to cow 2 starting the simulation in its first lactation.

The cumulative milk production of cow 1 was found to be 95,497.5 kg while the cumulative milk production of cow 2 was 96,292.8 kg. Both cow 1 and cow 2 went through 9 lactations and with an average of 10,000 kg milk per lactation they were both expected to produce around 90,000 kg milk. The difference in milk yield between cow 1 and cow 2 can be attributed to the variation in possible states at the beginning of the first lactation. This indicates that the model and simulation are working as intended.

In figure 5 the average nitrogen emission through feces and urine is shown for cow 1 and cow 2. Like in figure 3, the peaks in figure 5 can be compared with the peaks in figure 6, which shows nitrogen emission curves for the first 4 lactations of an average cow. Figure 5 follows the same trend as figure 3, where early lactations look very similar to the expected curve and show more variation in later lactations due to the overlap that is created.

The cumulative nitrogen emission for cow 1 was 1,606,860.0 g and 1,575,291.5 g for cow 2. When looking at the average daily nitrogen emission in figure 6, taking an average of 300 g for 4900 days, it sums up to a total of 1,470,000 g. Like with the milk production the model may overshoot on the cumulative phenotype output. For nitrogen emission this is hard to determine since collecting data on emissions per cow is difficult, but assuming that the formula's give an accurate representation of actual nitrogen emissions per cow, these results are expected.

Average milk production per day in simulation

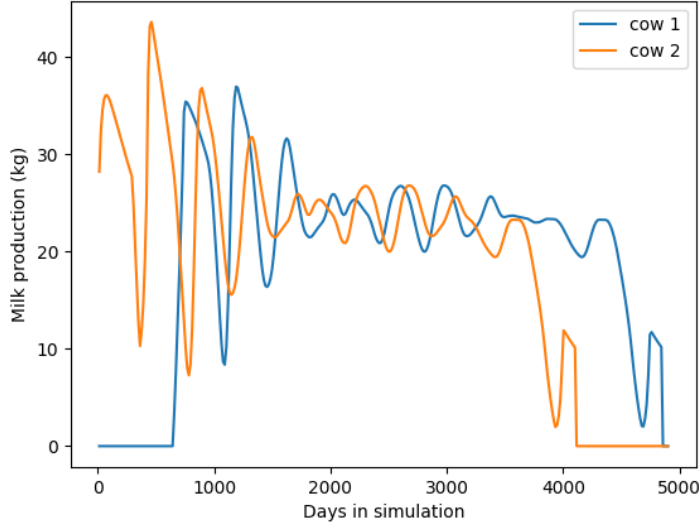


Figure 3: Average daily milk production during the simulation for a heifer (cow1) and a cow starting her first lactation (cow2).

Average lactation curves for the first 4 lactations

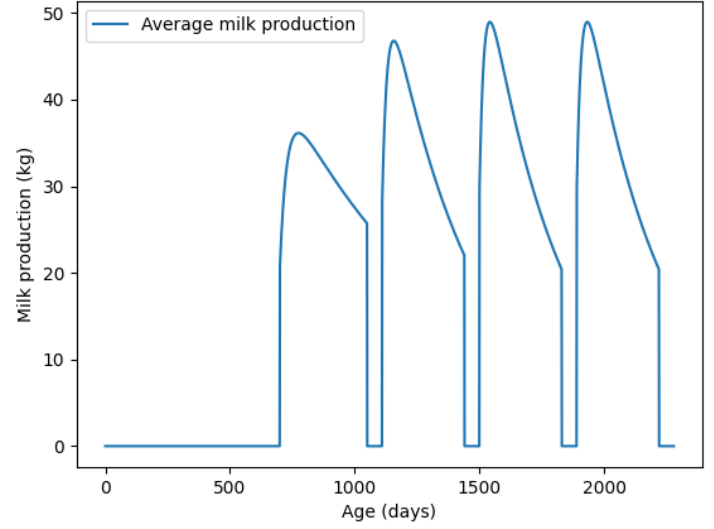


Figure 4: Daily milk production of the first 4 lactations of an average cow.

Average nitrogen emission per day in simulation

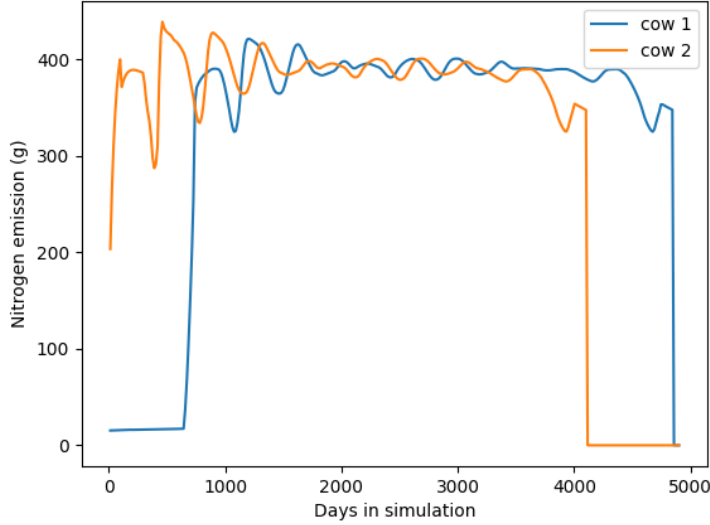


Figure 5: Average daily nitrogen emission through feces and urine during the simulation for a heifer (cow1) and a cow starting her first lactation (cow2).

Average nitrogen curves for the first 4 lactations

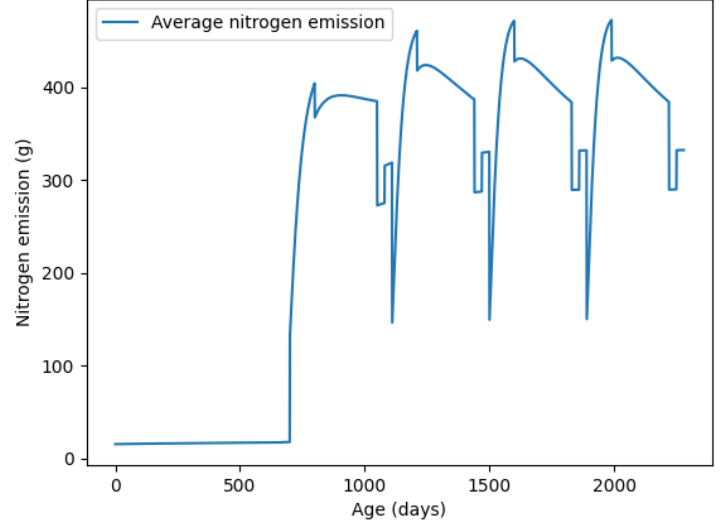


Figure 6: Daily nitrogen emission for the first 4 lactations of an average cow.

## Comparison of step size

Calculating the phenotype values for every day in the simulation is very computationally intensive the longer the simulation is, and therefore consumes a lot of time. When running the simulation, the parameter `interval` of the `state_vector_processor` function determines the interval in days for which a state vector is returned and phenotype values are calculated. If set to 1, the model will calculate the accumulated phenotype values for every day in the simulation. If the `interval` parameter is increased, the model will calculate phenotype values for a given vector and extrapolate these up until the next vector by multiplying them with the `interval` parameter. This will cause loss of accuracy, since extrapolating might overshoot or undershoot the actual values, but it can also drastically reduce computation time as the model has to process state vectors less frequently. In a comparison of performance for step size 7 and 14 to step size 1, computational time and phenotype variance were measured.

In figure 7 the computational time is shown for all 3 step sizes in seconds on a logarithmic scale, for 5 different simulation lengths. Step size 1 has a significantly longer run time then step size 7 and 14. In the longest simulation the difference between step size 14 and 1 is a factor 37.55. Exact numbers and conversions of time can be found in table 4 of the appendix.

Figures 8 and 9 show the percent variance in milk production and nitrogen emission respectively, comparing step size 7 and 14 to step size 1, for 5 different simulation lengths. In figure 8 both step size 7 and 14 have a slight overshoot compared to step size 1 for the simulation of 280 days and 560 days. This turns into an undershoot for the remaining simulations but stays very consistent at 99.75% for step size 7 and 99.5% for step size 14. In figure 9 step size 14 starts off with a relatively high overshoot of 101.65%, but this gradually converges with step size 7 and stabilises at 99.85%.

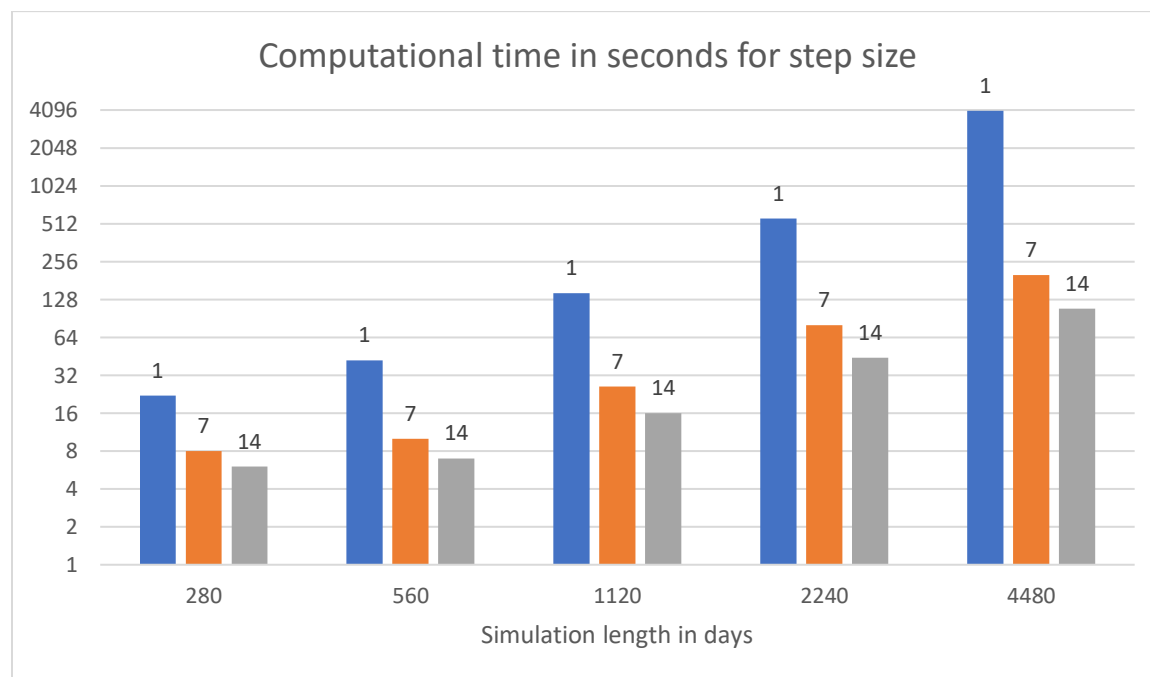


Figure 7: Comparison of computational time required to run a simulation of different lengths for step size 1, 7, and 14.

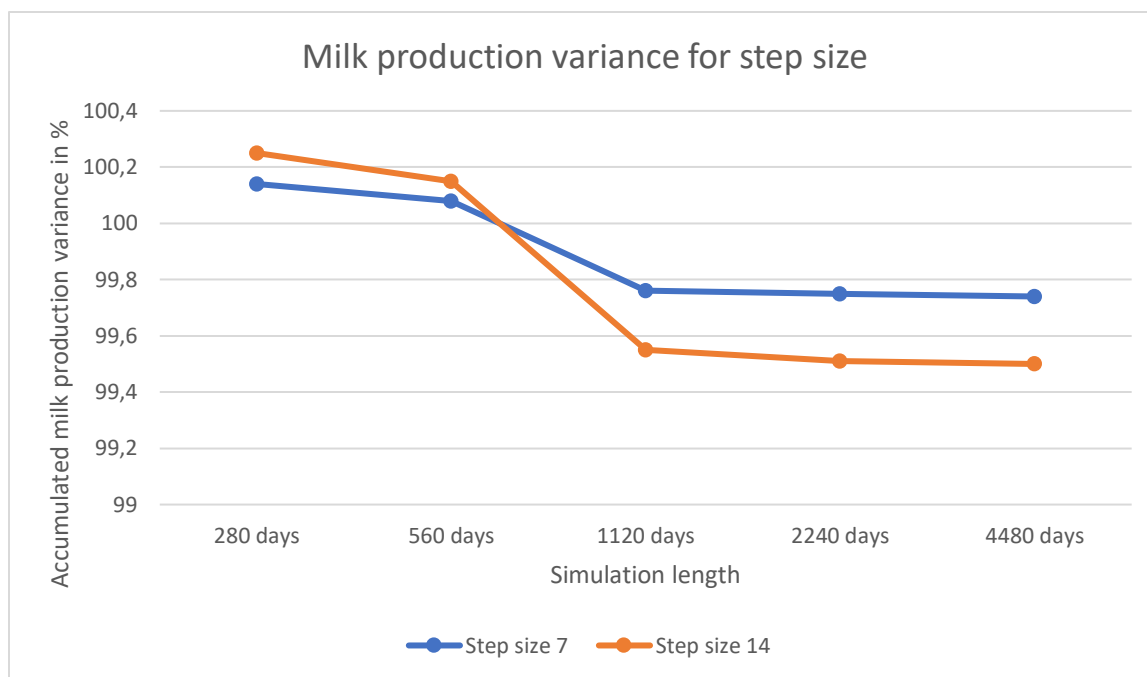


Figure 8: Variance in accumulated milk production for step size 7 and 14 when compared to step size 1, for different lengths of simulation.

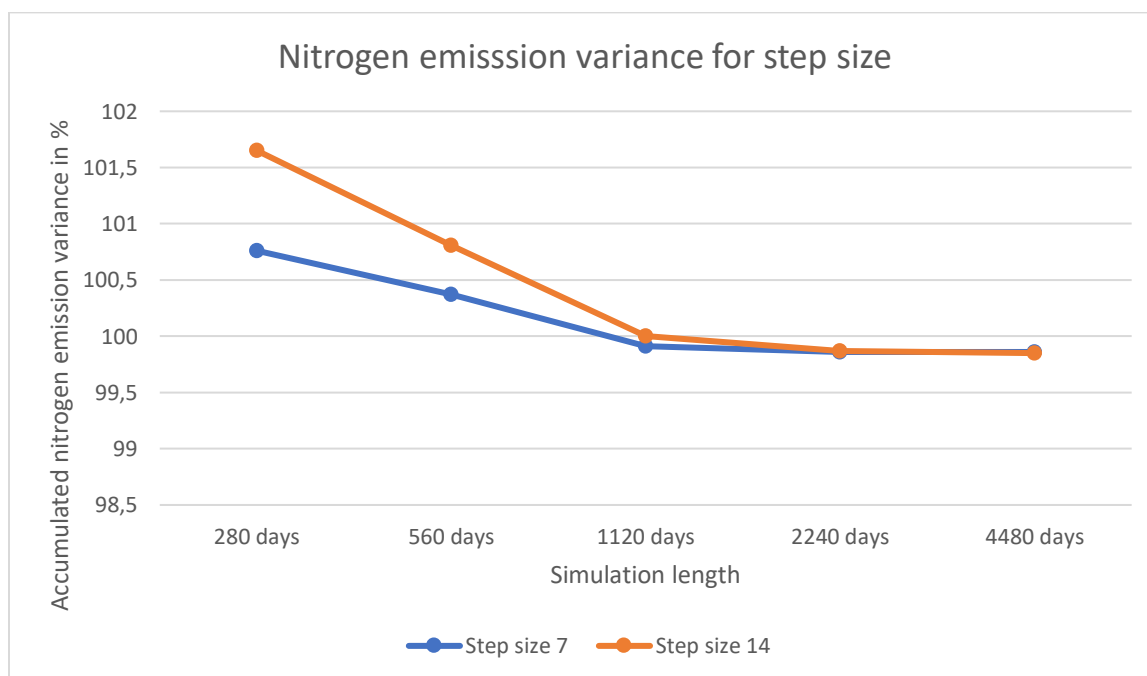


Figure 9: Variance in accumulated nitrogen emission for step size 7 and 14 when compared to step size 1, for different lengths of simulation.



## Discussion

### Decreasing computational time with the step size

As discussed in the results, the computational time required to perform a simulation can significantly be reduced by increasing the step size to reduce the number of intermediate calculations the model has to perform. This, in combination with the remarkably low variance in phenotype output, means that the step size can be increased to drastically reduce computational time while sacrificing almost no accuracy. It is therefore recommended to use a step size of 14 instead of 1. Larger step sizes may be used to reduce computational time further, although the effectiveness of a step size larger than 14 has not been tested, nor has the effect that this will have on the phenotype output. Therefore, further significant reductions in computational time and accurate results for step sizes larger than 14 cannot be guaranteed at this time.

### Cumulative phenotype output

Initially, the cumulative phenotype output was calculated by multiplying the phenotype output of every state that the cow could be in at that point in the simulation, with the probability of that state at that point, and taking the sum of this to get the weighted average output for that day. This was then summed for every day in the simulation. The problem arose that for long simulations the probabilities would get so small, that it effected the weighted average to the point that it did not accurately represent the daily phenotype output. Therefore, it was opted to take the sum of the average of the live states that the cow could be in at that point in the simulation. This does mean that the vector probabilities are not explicitly being used for calculating the cumulative output other than determining which states the model can be in at that point in the simulation. The overshoot for these phenotypes is somewhat significant and may be reduced by implementing a method that uses the vector probabilities. Further research should be conducted as to how this can be achieved.

### Milk production phenotype in state objects

The milk production is one of the components of the state objects. This was done to easily access the milk yield for any state to check if the milk production had fallen below the threshold, without having to recalculate the milk yield, saving time. However, when the standard deviation is increased for the MilkBot parameters, to simulate individual variation, milk yield may significantly vary between individuals. This would mean that transition matrices vary per cow, and therefore a new transition matrix would be required for every cow. Depending on the size of these matrices, the number of cows to be simulated, and the duration of the simulation, this may significantly increase computing time. During further development of this model, it is recommended that the milk yield is taken out of the state object.

### Cache for phenotype equations

When the parameters used by equations to calculate phenotypes do not change, caching equations will significantly reduce computing time as the model will not need to redo calculations it has done before. However, when variation is introduced by increasing the standard deviation of phenotype function parameters, this will lose effectiveness as calculations will have to be done with possibly only slight variation in the parameters. This will cause the computational time to increase again. This is something to keep in mind when altering the standard deviation of parameters.

### Replacement of a cow

In version 1.0 of this model, a cow will not be replaced by another during the simulation. This makes it so that accumulated phenotype values slow in growth over time as the probability that the cow is still alive decreases. The reason this implementation was chosen was that the equation used to get the body weight of the cow required the age of the cow in days. When dealing with a daily probability of death this meant that every day the cow could be replaced, and the age would reset. The problem was that the model also uses an insemination period, which makes it possible for a cow to be in the same state with a different age from lactation 1 onward. A solution that was proposed was to include the age in the state objects, so that the age could easily be tracked for every state. However, this causes the transition matrix to grow to extremely large proportions, to the point where it significantly increases computational time. Further development of this model should include to find a more manageable method of tracking the age of a cow in all states, so that the replacement implementation can be used without the need to include the age into the state objects.

### Daily probability of death

The model used a set daily probability of involuntary death. Having only one set probability may cause slight inaccuracies when simulating cows over an extended period of time. Further development of this model may look into the probabilities of involuntary death at different stages of the cow's life and implement this into the model to more accurately represent a real cow.

### Accessing intermediate state vector on GPU

During the simulation, vectors containing transition probabilities are returned at set intervals to calculate phenotype values. As of version 1.0, the model requires a data transfer of this vector from the GPU to CPU to perform these calculations when using the GPU. This data transfer puts a bottleneck on the computational time, especially when working with large transition matrices. This could be resolved by switching from NumPy to CuPy, which allows the model to access the intermediate state vector on the GPU. This could significantly reduce computational time, which allows for larger or more complex simulations.

### Validation of the model using case studies

Validation of the functionality of the model was done by comparing simulation results to known phenotype curves. To really evaluate the performance of this model, case studies may be done in which phenotype values are tracked for a group of cows during a certain period. Data on these cows collected at the beginning of the case study can then be used as parameters for the `DigitalCow` objects, after which a simulation of the same length as the period for which data was collected is done. The simulation results could then be compared to the collected data to assess predictive performance of the model.

### Addition of more phenotypes

Future work on this package could also include the addition of more phenotypes that can be simulated. Examples of this could be maintenance cost, phosphor emission, and methane emission. To do this, functions may be added to calculate these phenotypes for a given state and to process an intermediate state vector and return these results.

## Conclusion

The results from the simulation of the digital twin compared to the expected values a dairy cow would produce for the length of the simulation, demonstrate that the model can accurately predict milk production and nitrogen emission for a dairy cow. Furthermore, simulations can be performed quickly using a lower interval of intermediate calculations by increasing the step size. The model is easy to use, is well documented and build in such a way that it can be expanded upon. This study should be used as a foundation for a more complex model that can predict more phenotypes and take more of the complex variables into account. Improvements of current implemented methods, such as replacing NumPy with CuPy and accounting for a variable daily probability of involuntary death, can reduce computational time and create more accurate predictions.

## References

- Climate change 2021: The physical science basis : summary for policymakers : working group I contribution to the sixth Assessment report of the Intergovernmental Panel on Climate Change. (2021). IPCC.
- Committee on Nutrient Requirements of Dairy Cattle, Board on Agriculture and Natural Resources, Division on Earth and Life Studies, & National Academies of Sciences, Engineering, and Medicine. (2021). Nutrient Requirements of Dairy Cattle: Eighth Revised Edition (p. 25806). National Academies Press. <https://doi.org/10.17226/25806>
- De Vries, A. (2006). The DairyVIP Program to Evaluate the Consequences of Changes in Herd Management and Prices on Dairy Farms. Department of Animal Sciences, Florida Cooperative Extension Service, Institute of Food and Agricultural Sciences, University of Florida. <https://scholar.archive.org/work/hlr6l5f7sbfhpkcw4nxzrocp44/access/wayback/https://journals.flvc.org/edis/article/download/116301/114467/>
- Dutch dairy in Figures 2021. (2022). ZuivelNL. <https://www.zuivelnl.org/nieuws/engelstalige-versie-zuivel-in-cijfers-2021>
- Ehrlich, J. L. (2010). Quantifying shape of lactation curves, and benchmark curves for common dairy breeds and parities. *The Bovine Practitioner*, 88–95. <https://doi.org/10.21423/bovine-vol45no1p88-95>
- Giordano, J. O., Kalantari, A. S., Fricke, P. M., Wiltbank, M. C., & Cabrera, V. E. (2012). A daily herd Markov-chain model to study the reproductive and economic impact of reproductive programs combining timed artificial insemination and estrus detection. *Journal of Dairy Science*, 95(9), 5442–5460. <https://doi.org/10.3168/jds.2011-4972>
- Harris, C. R., Millman, K. J., Van Der Walt, S. J., Gommers, R., Virtanen, P., Cournapeau, D., Wieser, E., Taylor, J., Berg, S., Smith, N. J., Kern, R., Picus, M., Hoyer, S., Van Kerkwijk, M. H., Brett, M.,

- Haldane, A., Del Río, J. F., Wiebe, M., Peterson, P., ... Oliphant, T. E. (2020). Array programming with NumPy. *Nature*, 585(7825), 357–362. <https://doi.org/10.1038/s41586-020-2649-2>
- Hostens, M., Ehrlich, J., Van Ranst, B., & Opsomer, G. (2012). On-farm evaluation of the effect of metabolic diseases on the shape of the lactation curve in dairy cows through the MilkBot lactation model. *Journal of Dairy Science*, 95(6), 2988–3007. <https://doi.org/10.3168/jds.2011-4791>
- Miller, R. H., Kuhn, M. T., Norman, H. D., & Wright, J. R. (2008). Death Losses for Lactating Cows in Herds Enrolled in Dairy Herd Improvement Test Plans. *Journal of Dairy Science*, 91(9), 3710–3715. <https://doi.org/10.3168/jds.2007-0943>
- OECD-FAO Agricultural Outlook. (n.d.). [Data set]. OECD Publishing. <https://doi.org/10.1787/agr-outl-data-en>
- Okuta, R., Unno, Y., Nishino, D., Hido, S., & Loomis, C. (2017). CuPy: A NumPy-Compatible Library for NVIDIA GPU Calculations. *Proceedings of Workshop on Machine Learning Systems (LearningSys) in The Thirty-First Annual Conference on Neural Information Processing Systems (NIPS)*. [http://learningsys.org/nips17/assets/papers/paper\\_16.pdf](http://learningsys.org/nips17/assets/papers/paper_16.pdf)
- Poncheki, J. K., Canha, M. L. S., Viechnieski, S. L., & Almeida, R. D. (2015). Analysis of daily body weight of dairy cows in early lactation and associations with productive and reproductive performance. *Revista Brasileira de Zootecnia*, 44(5), 187–192. <https://doi.org/10.1590/S1806-92902015000500004>
- Roser, M., & Rodés-Guirao, L. (2013). Future Population Growth. *Our World in Data*.
- Transforming the livestock sector through the Sustainable Development Goals. (2018). FAO. <https://www.fao.org/3/CA1177EN/ca1177en.pdf>
- Tubiello, F. N., Karl, K., Flammini, A., Gütschow, J., Obli-Laryea, G., Conchedda, G., Pan, X., Qi, S. Y., Halldórudóttir Heiðarsdóttir, H., Wanner, N., Quadrelli, R., Rocha Souza, L., Benoit, P., Hayek,

M., Sandalow, D., Mencos Contreras, E., Rosenzweig, C., Rosero Moncayo, J., Conforti, P., & Torero, M. (2022). Pre- and post-production processes increasingly dominate greenhouse gas emissions from agri-food systems. *Earth System Science Data*, 14(4), 1795–1809.

<https://doi.org/10.5194/essd-14-1795-2022>

Van Arendonk, J. A. M. (1985). A model to estimate the performance, revenues and costs of dairy cows under different production and price situations. *Agricultural Systems*, 16(3), 157–189.

[https://doi.org/10.1016/0308-521X\(85\)90010-1](https://doi.org/10.1016/0308-521X(85)90010-1)

Zhang, Wang, Chang, Luo, Brito, Dong, Shi, Wang, Dong, & Liu. (2019). Mortality-Culling Rates of Dairy Calves and Replacement Heifers and Its Risk Factors in Holstein Cattle. *Animals*, 9(10), 730.

<https://doi.org/10.3390/ani9100730>

## Appendix

Table 4: Results for the comparison of computational time and phenotype variance when running simulations for step size 1, 7, and 14.

Days simulated	Steps	Time sec	Time	Time difference factor	Milk (kg)	Milk variance	N (g)	N variance
280	1	22	00:00:21.8		8901.9		98117.7	
	7	8	00:00:07.7	2.75	8914.7	+0.14%	98863.3	+0.76%
	14	6	00:00:06.2	3.67	8925.0	+0.25%	99736.1	+1.65%
560	1	42	00:00:42.2		16049.1		195605.7	
	7	10	00:00:10.2	4.20	16062.3	+0.08%	196330.8	+0.37%
	14	7	00:00:07.4	6.00	16072.9	+0.15%	197181.9	+0.81%
1120	1	144	00:02:24.1		25263.7		355422.0	
	7	26	00:00:26.2	5.54	25203.3	-0.24%	355115.7	-0.09%
	14	16	00:00:15.6	9.00	25149.5	-0.45%	355430.1	+0.00%
2240	1	561	00:09:21.3		28917.6		411136.4	
	7	80	00:01:19.7	7.01	28844.4	-0.25%	410571.0	-0.14%
	14	44	00:00:44.4	12.75	28775.7	-0.49%	410583.4	-0.13%
4480	1	4055	01:14:14.9		29436.8		419304.5	
	7	200	00:03:20.4	20.28	29361.2	-0.26%	418697.2	-0.14%
	14	108	00:01:47.5	37.55	29289.7	-0.50%	418661.0	-0.15%

Table 5: Python packages used to create the cow-builder and run simulations, with versions and usage descriptions.

Package	Version	Required	Used for
chain-simulator	0.3.2	yes	Creating a transition matrix and simulating transitions
numpy	1.24.3	yes	Numpy arrays and random samples
jupyter	1.0.0	no	Running Jupyter notebooks
setuptools	65.5.1	no	Building the package
sphinx	7.0.1	no	Documentation
furo	2023.5.20	no	Documentation layout
matplotlib	3.7.1	no	Creating plots

Table 6: Functional requirements with MSCW description and priority levels High-Middle-Low.

ID	Description	Priority	Moscow
1	The model can create all possible new states for a specific instance. The number of states can be controlled, with parameters	High	M
2	The model can calculate the probability for changing from one state to another	High	M
3	The model can store important instance variables and manipulate them	High	M
4	The model can calculate nitrogen emission from current or simulated instance values	High	M
5	The model can calculate milk production from current or simulated instance values	Middle	M
6	The model can calculate DMI from current or simulated instance values	Middle	S
7	The model can calculate methane emission from current or simulated instance values	Middle	C

Table 7: Non-functional requirements with MSCW description and priority levels High-Middle-Low.

ID	Description	Priority	Moscow
1	The model is written in Python	High	M
2	The model is developed with scalability of functionality in mind, meaning that extra features and functionality can be added easily	High	M
3	The model is developed with scalability of usage in mind, meaning that the model can be easily applied to large groups	High	M
4	The model is tested with unit-tests	Low	S
5	The model is well documented, having PEP8 Docstrings to support the code as well as a software document explaining functionality in detail.	High	S
6	Usage examples of the model are provided in the documentation or code	Middle	C
7	The model uses little to no dependencies, to easily maintain functionality over long periods of time	High	S