A new standard model for milk yield in dairy cows based on udder physiology at the milking-session level

Patrick Gasqui* & Jean-Marie Trommenschlager

Supplementary Information

Supplementary Figure 1. Distribution of between-session duration. Histogram of interval duration (black bars), with a mean of 10 and 14 hours for the morning and evening milking sessions, respectively (dashed green lines). The adjusted distribution of interval duration (solid blue curve) follows a normal distribution, and the morning and evening milking sessions have the same standard deviation of 0.50.
Supplementary Figure 2. Session-specific milk yield from the simulation and the model. a, b, simulated yield (n=1,000; black line), observed yield (green line), and model-estimated yield±95% CI (solid and dashed red lines, respectively) for the morning (a) and evening (b) milking sessions. c, plot of the total session-specific variance in milk yield found using the model versus the simulation (dashed red best-fit line).

Supplementary Figure 3. Daily milk yield from the simulation and the model. a, simulated yield [n=1000] (black line), observed yield (green line), and model-estimated yield±95 CI (solid and dashed red lines, respectively). b, plot of mean daily milk yield found using the model versus the simulation (dashed red best-fit line). c, plot of the total daily variance in milk yield found using the model versus the simulation (dashed red best-fit line).
Supplementary Figure 4. Relative contribution of variance components to total variance over the lactation period. Session-specific estimates (% provided is the estimate for whole lactation period): proportion of activated alveoli (62.1%; red line), interval duration (18.8%; magenta line), measurement error (7.9%; black line), $\beta$ (6.1%; green line), $\pi$ (2.6%; blue line); and milk carryover (2.5%; cyan line).
Supplementary Figure 5. Observed versus estimated milk yield full days when precise interval duration was unknown. Observed yield is represented by a solid black line. Model-estimated yield and the 95% CI are represented by green lines (solid and dashed, respectively). The estimated yield from Wood’s model and the 95% CI are represented by red lines (solid and dashed, respectively).
Supplementary Figure 6. Image illustrating the simulation used to estimate parameter influence in the milk-yield model.
Note: “Parameter estimation using the nlm function in R.”

The R function nlm carries out the non-linear minimisation of target functions, and we used it to estimate all the model parameters at once. Based on the output, we can see that this process seemed to run smoothly.

```R
resNLM <- nlm(fCMVMPUtilTempsEV, pliniEVG, xDlTemps, hessian = TRUE, iterlim = 400)
resNLM
```

```
$minimum
 [1] 1841.758

$estimate
[1] 4.26236099 -0.24851020 10.09942585 1.16457498 2.64294427 -0.61988279 0.24482785 -0.06540155 -0.31801924

$gradient
[1] 1.947076e-06 -1.001326e-05 1.076399e-05 -1.922155e-06 -1.184601e-06 -3.002382e-07 -3.722125e-06 -2.923002e-05 1.542517e-05

$Hessian

[1,] 1.128383767 162.9949969 17.7052240 6.021159 237.176031 0.5471389 -7.3233512e-01 -8.099211e+00 -1.317717e+00
[2,] 1.162.9949969 431.9892014 0.2630941 16.467608 170.429226 -1.9909812 -5.430297e+00 -1.779313e+00 -2.898310e+00
[3,] 1.17.7052240 0.2630941 105.404257 13.788833 -69.720995 45 83.949642 2.004714e+00 -3.518732e+00 -9.157320e+00
[4,] 1.6.0211593 16.467608 164.260882 -13.788832 141.939764 10.179264 -21.626761 2.190712e+00 -7.488163e+00 -9.908817e+00
[5,] 1.237.176031 170.429226 69.720995 10.179264 79.66408356 1.417795 1.899545e+00 2.702644e+04 4.227002e+02
[6,] 5.0471389 -1.9909812 83.949642 1.899545 21.626761 356.141779 130.188228 7.449988 0.1 7.913040e+01 2.805268e+00
[7,] -0.7323512 -5.430297 -2.004714 -3.518732 1.499545 -0.744989 5.453515 1.01 5.385095e+00 -2.737373e+05
[8,] 80.9922150 177.9130798 351.8729095 70.848629 270.2644065 79.33046 5.385095e+00 2.677550e+04 6.264531e+00
[9,] -1.3177176 -2.898310 417.200218 2.805268 2.273737e+05 6.264531e+00 4.174649e+02
[10,] -6.5089853 -14.318470 -470.6328809 5.088215 2112.717288 15.4587042 -2.273737e+05 -2.737373e+05 -6.25813e+00
[11,] -4.0749252 -8.962774 -312.2035878 6.079477 1325.320906 10.136773 -2.273737e+05 -2.273737e+05 0.0000000e+00

$code
[1] 1

$iterations
[1] 78
```
1 – The output code was 1, which, according to the nlm documentation, is the best value you can obtain.

```
| Code | Value |
|------|-------|
| 1    | 1     |
```

"nlm R documentation"
"code : an integer indicating why the optimization process terminated
1 : relative gradient is close to zero, current iterate is probably solution.
2 : successive iterates within tolerance, current iterate is probably solution.
3 : last global step failed to locate a point lower than estimate. Either estimate is an approximate local minimum of the function or steptol is too small.
4 : iteration limit exceeded.
5 : maximum step size stepmax exceeded five consecutive times. Either the function is unbounded below, becomes asymptotic to a finite value from above in some direction or stepmax is too small."

2 – The number of iterations was 78, which is not very high, given that we set the maximum number of iterations to 400 (via interlim); the default value is 100.

```
| Iterations | Value |
|------------|-------|
|            | 78    |
```

It is perhaps worth noting that nlm uses a Gauss-Newton-type algorithm, which is not necessarily optimal in complex non-linear situations. On the one hand, we would have preferred a Gauss-Marquart-type algorithm, which is more optimised for cases such as ours and more rapidly converges towards a solution (i.e., requires far fewer iterations). However, no such option exists for nlm. On the other hand, we did want to use a general non-linear minisation function such as nlm to demonstrate that parameter estimation can be efficacious and robust when commonly available procedures are used.

3 – In nlm, numerical derivatives are used for the gradient vector and the Hessian matrix. Algorithm speed increases when the vector and the matrix are explicitly calculated. The process yields a Hessian matrix whose inverse is easy to obtain using the solve function. This shows that all the parameters could be estimated. From the Hessian matrix, we obtained a variance-covariance matrix for all of the parameters; the main diagonal was the linear vector of the variances.

```
matHes <- resNLM$hessian
matVCW <- solve(matHes)
varParEl <- diag(matVCW)
```

The fact that we obtained reasonable and coherent values for these variance estimators is another sign that parameter estimation ran smoothly. We can thus feel confident that all of the parameters could be estimated.
To show that the values presented in the final versions of the tables (1, 2, 3, and 4) correspond to what was obtained using the nlm function, we present the output (resNLM – see above).

\[
\text{espParB} <- c(\exp(\text{resNLM}\$estimate[1:3]), \exp(\text{resNLM}\$estimate[6])) \\
> \text{espParB} \\
\begin{bmatrix}
7.097736e+01 & 6.448866e-03 & 4.497401e-05 & 5.380075e-01
\end{bmatrix}
\]

Delta time $\delta$ (hour) = 70.98

Activation rate $\lambda_A$ (per hour) = $6.4 \times 10^{-4}$

Inactivation rate $\lambda_R$ (per hour) = $4.5 \times 10^{-4}$

Inflation variance factor $\phi = 0.54$

\[
\text{espParT} <- c(\exp(-\exp(\text{resNLM}\$estimate[4])), \exp(-\exp(\text{resNLM}\$estimate[7:(7+\text{nbTV}-1)]))) \\
> \text{espParT} \\
\begin{bmatrix}
0.04505763 & 0.27876075
\end{bmatrix}
\]

Retention rate $\pi(k)$ “for 33 and 361” = 0.28

Retention rate $\pi(k)$ “for others” = min($\pi(k)$) = 0.04

\[
\text{espParP} <- c(\exp(\text{resNLM}\$estimate[5]), \exp(\text{resNLM}\$estimate[5]) \times \exp(\text{resNLM}\$estimate[(7+\text{nbTV}):\text{length(\text{resNLM}\$estimate)])}) \\
> \text{espParP} \\
\begin{bmatrix}
14.054523 & 13.164749 & 10.225913 & 11.622089 & 9.012371
\end{bmatrix}
\]

Secretion rate PL(k) estimate (in hg per hour) of table 4 = 14.1 13.2 11.6 10.2 9.0

Secretion rate max (in hg per hour) = max(PL(k)) = 14.1