COMPUTER SIMULATION MODELS OF THE PORCUPINE CARIBOU HERD: II. GROWTH

F.W. Hovey
L.L. Kremsater
R.G. White
D.E. Russell
F.L. Bunnell

TECHNICAL REPORT SERIES No. 54
Pacific and Yukon Region 1989
Canadian Wildlife Service

Environment Canada | Environnement Canada
Canadian Wildlife Service | Service canadien de la faune

Canada

TECHNICAL REPORT SERIES
CANADIAN WILDLIFE SERVICE

This series of reports, established in 1986, contains technical and scientific information from projects of the Canadian Wildlife Service. The reports are intended to make available material that either is of interest to a limited audience or is too extensive to be accommodated in scientific journals or in existing CWS series.

Demand for these Technical Reports is usually confined to specialists in the fields concerned. Consequently, they are produced regionally and in small quantities; they can be obtained only from the address given on the back of the title page. However, they are numbered nationally. The recommended citation appears on the title page.

Technical Reports are available in CWS libraries and are listed with the DOBIS system in major scientific libraries across Canada. They are printed in the official language chosen by the author to meet the language preference of the likely audience. **To determine whether there is significant demand for making the reports available in the second official language, CWS invites users to specify their official language preference. Requests for Technical Reports in the second official language should be sent to the address on the back of the title page.**

SÉRIE DE RAPPORTS TECHNIQUES
DU SERVICE CANADIEN DE LA FAUNE

Cette série de rapports donnant des informations scientifiques et techniques sur les projets du Service canadien de la faune (SCF) a démarré en 1986. L'objet de ces rapports est de promouvoir la diffusion d'études s'adressant à un public restreint ou trop volumineuses pour paraître dans une revue scientifique ou l'une des séries du SCF.

Ordinairement, seuls les spécialistes des sujets traités demandent ces rapports techniques. Ces documents ne sont donc produits qu'à l'échelon régional et en quantités limitées; ils ne peuvent être obtenus qu'à l'adresse figurant au dos de la page titre. Cependant, leur numérotage est effectué à l'échelle nationale. La citation recommandée apparaît à la page titre.

Ces rapports se trouvent dans les bibliothèques du SCF et figurent aussi dans les listes du système de référence DOBIS utilisé dans les principales bibliothèques scientifiques du Canada. Ils sont publiés dans la langue officielle choisie par l'auteur en fonction du public visé. **En vue de déterminer si la demande est suffisamment importante pour produire ces rapports dans la deuxième langue officielle, le SCF invite les usagers à lui indiquer leur langue officielle préférée. Il faut envoyer les demandes de rapports techniques dans la deuxième langue officielle à l'adresse indiquée au verso de la page titre.**

COMPUTER SIMULATION MODELS OF THE PORCUPINE CARIBOU HERD: II. GROWTH

F.W. Hovey[1]
L.L. Kremsater[1]
R.G. White[3]
D.E. Russell[2]
F.L. Bunnell[1]

Technical Report Series No. 54
Pacific and Yukon Region 1989
Canadian Wildlife Service

This series may be cited as:

Hovey, F.W., L.L. Kremsater, R.G. White, D.E. Russell, and F.L. Bunnell. 1989. Computer simulation models of the Porcupine caribou herd: II. Growth. Technical Report Series No. 54. Canadian Wildlife Service, Pacific and Yukon Region, British Columbia.

1 Faculty of Forestry, University of British Columbia, Vancouver, B.C., V6T 1W5
3 Institute of Arctic Biology, University of Alaska, Fairbanks, Alaska, 99775
2 Canadian Wildlife Service, 204 Range Rd., Rm. 202, Whitehorse, Yukon, Y1A 3V1

Published by Authority of the
Minister of Environment
Canadian Wildlife Service

Minister of Supply and Services Canada 1989
Catalogue No. CW69-5/54E
ISBN 0-662-16693-0
ISSN 0831-6481

Copies may be obtained from:
Canadian Wildlife Service,
Pacific and Yukon Region
P.O. Box 340,
Delta, British Columbia,
Canada V4K 3Y3

ABSTRACT

The growth of an individual caribou is simulated on a daily time period over 15 life cycle periods of the Porcupine caribou herd. In this model input variables (metabolizable energy intake [from ENERGY model], activity budgets, snow depths, etc.) are incorporated to simulate the growth of a female and her calf. The present model is one of three models, ENERGY, GROWTH and HARVEST, that were developed at the request of the Porcupine Caribou Technical Committee. The models run on IBM compatable microcomputers that have graphics capability. The supervisor program, Microsimcon, is incorporated into each model to assist in exercising the models.

RÉSUMÉ

La croissance d'une caribou individuel est simulée sur une période de temps quotidient pendant 15 périodes du cycle vital du troupeau de caribous Porcupine. Dans ce modèle, les variables d'entrée (apport en énergie métabolisable [tiré du modèle ÉNERGIE], les budgets d'activités, la profondeur de la neige, etc.) sont incorporées pour simuler la croissance d'une femelle et de son veau. Ce modèle fait partie de trois modèles (ÉNERGIE, CROISSANCE et RÉCOLTE) qui ont été mis au point à la demande de Comité technique du caribou Porcupine. Les modèles sont utilisables sur les micro-ordinateurs ayant une compatibilité IBM et une capacité inforgraphique. Le programme superviseur, Microsimcon, est incorporé dans chaque modèle et facilite l'application et la vérification des modèles.

INTRODUCTION

Research and monitoring of the Porcupine caribou herd have been conducted almost continuously since the early 1970's. As more information is gathered it becomes increasingly difficult to integrate and apply that information to complex questions regarding effects on the herd from human activities and developments. The Porcupine Caribou Technical Committee, therefore, requested that individuals involved in research and management of the herd develop computer simulation models to aid in evaluating the present data, to help guide future research, and to provide some insights into the potential impact of alternate development scenarios. To this end the Canadian Wildlife Service contracted a group from the Faculty of Forestry, University of British Columbia, to help facilitate a number of workshops and to program the agreed upon models. From those discussions, three models were proposed. By order of increasing iteration time these models are: an ENERGY model which simulates the energetic relations of an individual and predicts the metabolizable energy intake (MEI) on a daily basis; a GROWTH model that incorporates the resultant MEI and projects the weight gain and loss throughout the year; and a HARVEST model that simulates the demographics of the herd over a number of years.

The models incorporate a simulation supervisor program that was adapted to microcomputers (Microsimcon), allowing efficient and user-friendly access while exercising the models. All coding is done in Basic using a "Quick Basic" compiler. All users of the models will require an IBM compatible computer with graphics capability.

This publication represents Version 1 of the models. Updates, executable files and other information on the model are available from D.E. Russell, Canadian Wildlife Service, Whitehorse.

ACKNOWLEDGEMENTS

Funding for this project was primarily from the Canadian Northern Oil and Gas Action Program, in the initial year through Environment Canada, Western and Northrn Region and in the second year through the Department of Indian Affairs and Northern Development, Northern Affairs Program. We especially thank Bill Brakel and Donna Stewart of those agencies, respectively, for their continued support. Funding was also received from the Canadian Wildlife Service, Pacific and Yukon Region. The list of authors reflect in large part the concentration of work and ideas that went into each model. Many individuals, however, contributed to the final product. Steve Fancy and Ann Allaye-Chan contributed significantly to the GROWTH model, and Laurie Kremsater to the HARVEST model. Many of the data that are incorporated in the models are unpublished and we thank those individuals for their contributions.

CARIBOU GROWTH MODEL

PURPOSE

The broad purpose of the growth model is to evaluate effects of changes in seasonal activity budgets and metabolizable energy intake on the energetics and reproductive status of a female caribou.

The model has two specific objectives:

(1) To evaluate the impact of changing activity costs, maintenance costs, and metabolizable energy intake on the cow's energy balance and subsequent growth.

(2) To evaluate effects of the cow's energy balance on the growth of her fetus during pregnancy and her calf during lactation.

GENERAL DESCRIPTION

The growth model computes the energy status of a female caribou on a daily timestep. The calculations it uses can be grouped into four categories: 1) energy intake, 2) energetic requirements of maintenance and activity, 3) energetic requirements of gestation and lactation, and 4) energetic requirements of growth and fattening (Fig. 1).

Currently, the model calculates energy intake from a modified sine curve. Thus, intake is not linked to the cow's food resources. The model, however, is designed to be compatible with the output and structure of the ENERGY model. The link to food resources can be made once the two models have been revised.

Energy requirements for maintenance follow Klieber's equation. The requirements for activity are derived from the cow's daily activity budget and activity costs derived from the literature, using those specifically for caribou whenever possible. The model treats 6 activities: foraging, lying, standing, walking, running, and in winter, cratering. The daily cost of each activity is calculated by determining the number of hours the cow spends performing the activity multiplied by the activity's energetic cost per hour. The sum of the costs of the 6 activities the total daily activity cost.

Costs of gestation and lactation are based on a rather novel approach. The energy associated with each phase of reproduction is calculated from daily target rates of output (for gestation the output is fetus weight and for lactation the output is milk production). Those rates are adjusted downwards by the cow's energy status, i.e. MEI and her fat and lean tissue reserves.

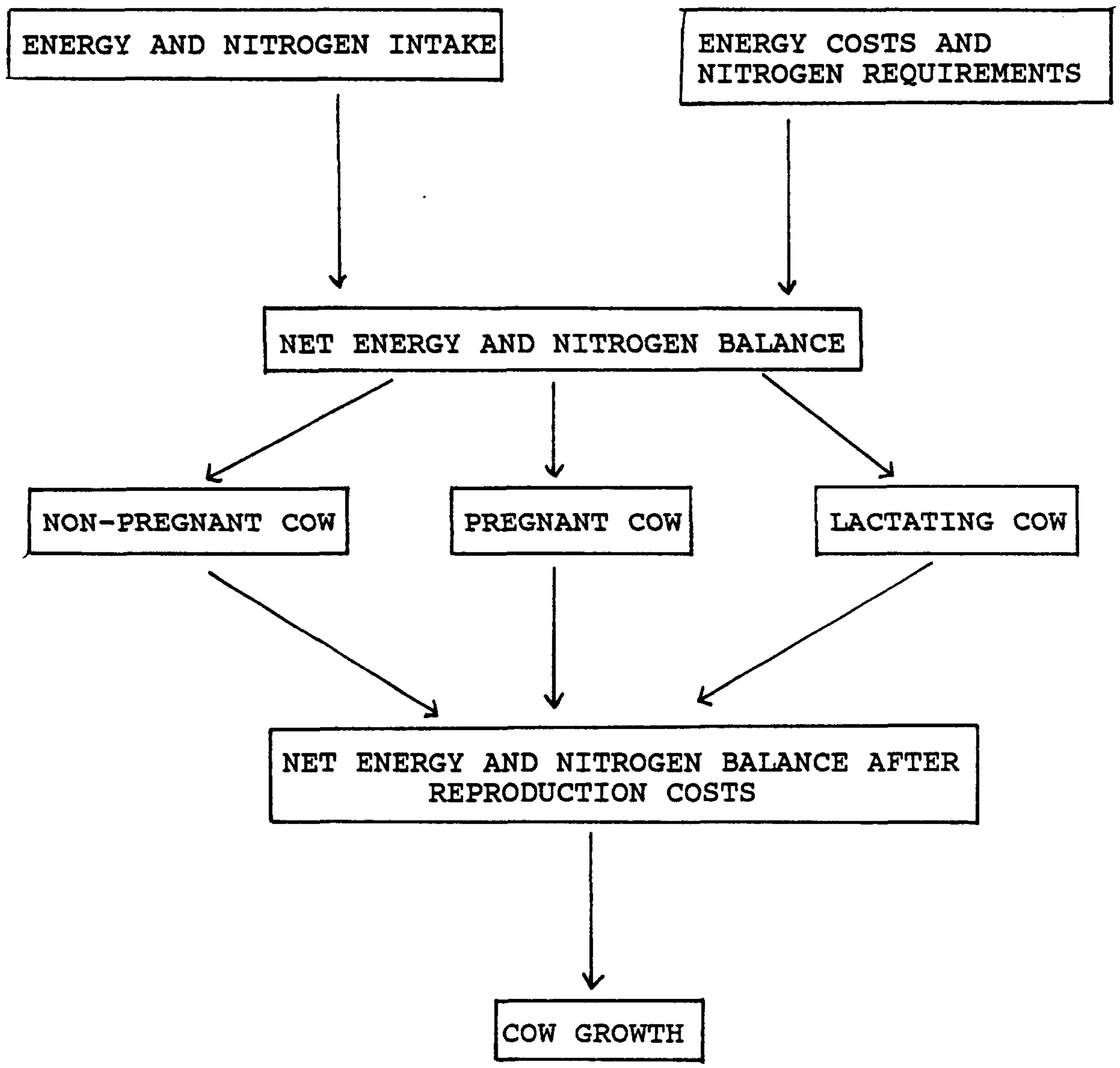

Figure 1. Flowchart of GROWTH model

When the female has a surplus of energy, the reproductive outputs are at the target levels. However, when she does not have enough energy to meet the target demands, the output is reduced in direct proportion to the difference between what she needs for maximum production and what she actually has available (Figs 2 and 3).

Changes in the cow's body composition due to catabolism and anabolism of fat and lean tissue stores follow, for the most part, the methods of Fancy (1986).

STEPPING THROUGH THE MODEL

A. Calculating MEI

After the appropriate variables have been initialized (see App. 1), the model begins each iteration of the simulation by calculating the cow's metabolizable energy intake (MEI) using a modified sine function. Potentially, MEI could be passed from the Energy model to the Growth model. To facilitate understanding and review of the models, we have avoided linking the models.

MEI = (EIMIN + (EIMAX - EIMIN) * ((1 + SIN(A*DAY+B))/2)) * MNMEI * WT75 (1)

where MEI = metabolizable energy intake ($kJ \cdot day^{-1}$),
EIMIN = variable controlling the minimum value of the sine function,
EIMAX = variable controlling the maximum value of the sine function,
A = factor that affects the period and horizontal shift of the sine function,
B = factor that affects the horizontal shift of the sine function,
DAY = day of simulation,
MNMEI = maximum MEI value ($kJ \cdot day^{-1}$), and
WT75 = cow's metabolic weight ($kg^{0.75}$).

B. Maintenance and Activity Costs

The model calculates heat production (HP, $kJ \cdot day^{-1}$), cost of maintenance ($kJ \cdot day^{-1}$ Hudson and Christopherson 1986), and heat increment (HI, $kJ \cdot day^{-1}$; Fancy 1986) as,

HP = 293 * WT75 (2)

MAINCST = HP / EFMAIN (3)

where EFMAIN = the efficiency at which metabolizable energy can be used for energy maintenance (Hudson and White 1986),
= (.35 * QM) + 0.503 (where QM is the metabolizability coefficient of the gross energy of diet (ARC 1980)

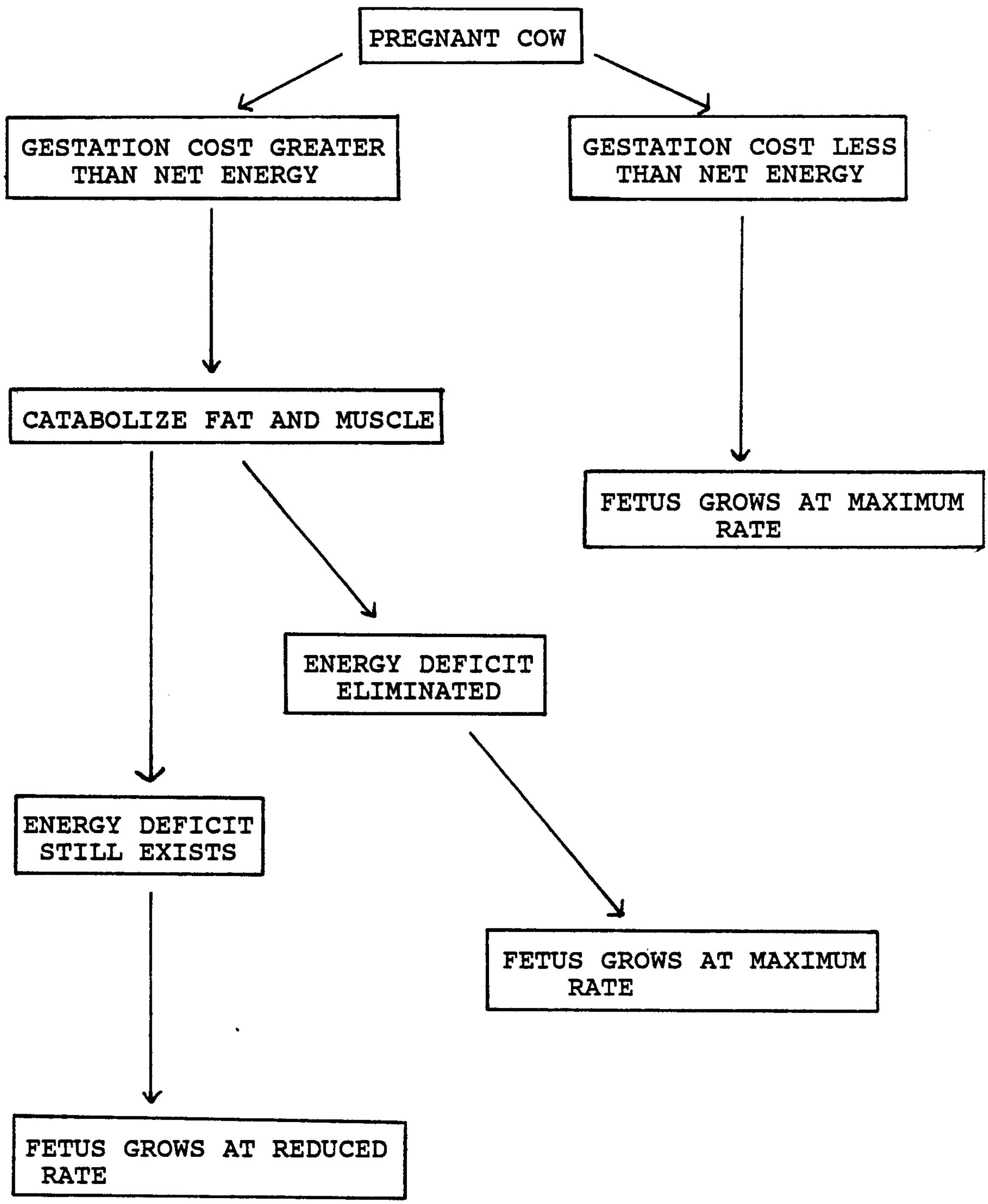

Figure 2. Calculation of gestation status.

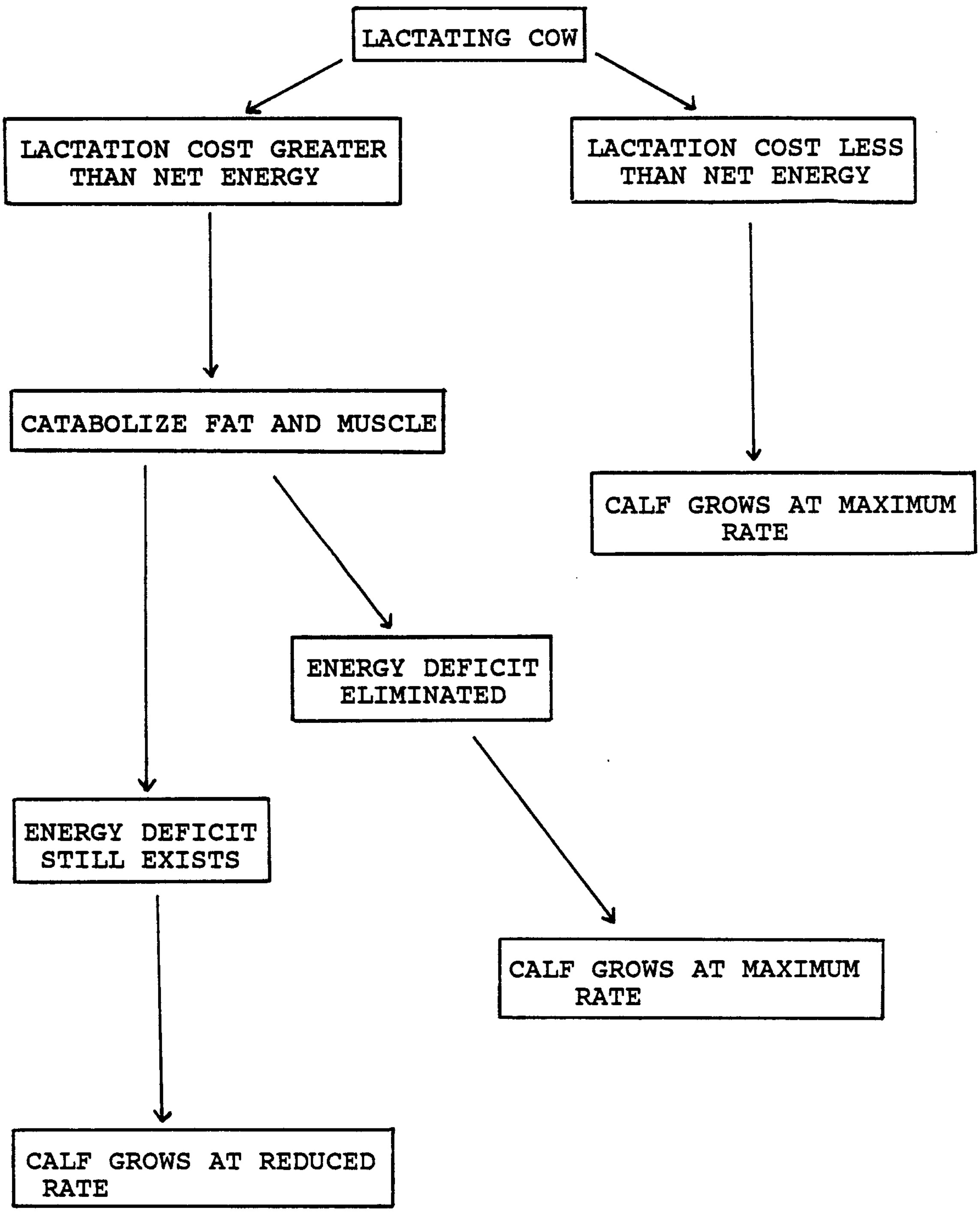

Figure 3. Calculation of lactation status

$$HI = (1 - EFMAIN) * MEI \quad (4)$$

The energetic costs of standing, walking, feeding, running, and cratering are determined from the cow's activity budget and body weight.

Because the activity budget is on a seasonal basis, the model linearly interpolates each activity between seasons to provide daily activity budgets. These interpolated proportions rarely add to one when summed across activities. To correct this problem, the model normalizes the activity budget by dividing each activity's proportion of the budget by the sum of the proportions of all the activities (PTOTAL).

When caribou are foraging in winter, not all of the time spent foraging is spent actually consuming food. Often, the caribou must paw through snow to uncover forage buried beneath. The model accounts for time spent cratering (pawing) and eating while foraging by separating the foraging activity into two components: 1) eating intensity, which is the proportion of the foraging period spent consuming food, and 2) pawing intensity, which is the proportion of the foraging period spent cratering for forage.

Before calculating the costs of each activity, the model calculates the snow depth (SNODEP) and the cow's sinking depth in the snow (SINKDEP) to determine the added cost of locomotion in snow (SNOWX). Snow depth is calculated by linearly interpolating seasonal-specific snow depths to provide daily values (DSNODEP). Sinking depth (cm) is calculated as,

$$SINKDEP = SDPROP * DSNODEP \quad (5)$$

where: SDPROP = proportion of the snow depth to which the cow sinks.
DSNODEP = daily snow depth (cm)

The added cost of locomotion in snow (SNOWX) is determined from the equation (Fancy 1986):

$$SNOWX = 2.41623 * e^{(0.0635 * SINKDEP * 1.587)}/1000 + 1 \quad (6)$$

The basic form of the equation calculating the daily energetic cost of each activity is:

$$CSTNRG(ACTIVITY) = COST(ACTIVITY) * ACTIVE(ACTIVITY)/PTOTAL * COWWT * 24 \quad (7)$$

where CSTNRG(ACTIVITY) = daily energetic cost of each activity ($kJ \cdot day^{-1}$),
COST(ACTIVITY) = hourly energetic cost of each activity ($kJ \cdot Kg^{-1} \cdot hour^{-1}$),
ACTIVE(ACTIVITY) = proportion of each day performing activity,
PTOTAL = sum of proportion of all daily activities, and

COWWT = weight of female caribou (kg).
24-h·day^{-1}

To account for cost of locomotion in snow, the daily activity costs of walking and running are multiplied by the factor, SNOWX defined earlier.

The model sums the daily costs of all activities to produce the total daily activity cost, EACTIVE (kJ·day^{-1}). Using EACTIVE, the model calculates the net energy (NETNRG, kJ·day^{-1}) available for growth and reproduction (gestation or lactation) as,

$$NETNRG = MEI - (EACTIVE + HP)/EFMAIN \quad (8)$$

C. Calculating Reproduction Costs

1) Gestation

The model determines if the cow is pregnant by stepping through a series of decisions (Boolean logic operations). First, it compares the variable DAY (representing the model's current iteration) with the variable CDAY (representing the Julian date of conception). If DAY is greater than or equal to CDAY, then the model checks if the cow has previously aborted her fetus (monitored by the logical variable, ABORT$). If the cow has not aborted her fetus, the model considers her to be pregnant. Because the model operates on a Julian day timestep, the first part of the decision process (comparing CDAY to DAY) may fail when the model is simulating dates from January 1 to date of parturition (e.g., January 1 = 1 < CDAY). To correct this problem, the model uses an additional logical variable, PREG$, to monitor the cow's pregnancy status. PREG$ is given a "YES" value when CDAY is greater than or equal to DAY and retains its "yes" value throughout the gestation period unless the fetus is aborted.

If the logic operations described above indicate that the cow is pregnant, the model calculates the target or ideal weight of the fetus (TFETWT, kg) according to the number of days she has been pregnant (DPREG) (corrected from Fancy 1986):

If the cow is less than or equal to 76 days pregnant, then

$$TFETWT = (0.00036 * DPREG^3) + (0.053*DPREG^2) - (1.58 * DPREG) - 0.000096 \quad (9)$$

If the cow is more than 76 days pregnant, then

$$TFETWT = (6.05254E^{-08} * DPREG^4) - (3.06828E^{-05} * DPREG^3) + 0.05719*DPREG^2) - (0.44743*DPREG) + 12.43291 \quad (10)$$

where: E = * 10

Equations (8) and (9) were developed for reindeer. To convert the results to caribou, the model adjusts target fetus weight by the equation (Fancy 1986):

$$TFETWT = (TFETWT * (BIRWT / 5.89)) / 1000 \quad (11)$$

where BIRWT = target birth weight (kg) set by the user, and
5.89 = the predicted birth weight of reindeer assuming a gestation length of 220 days.

Based on the target weight of the fetus, the model calculates the energetic cost of gestation for that day of pregnancy (EGEST, $kJ \cdot day^{-1}$; Robbins 1983) as:

$$EGEST = 2.4E^{-07} * (DPREG / GESLEN * 100)^{3.13}) * MAINCST \qquad (12)$$

where GESLEN = gestation length (days).

From the energetic cost of gestation, the model calculates the cost of maintaining the conceptus (CONCCOST, kJ·per kg fetus weight per day) by the equation:

$$CONCCOST = EGEST / TFETWT \qquad (13)$$

The model's next step is to use the target weight to determine the target daily growth rate of the fetus (TGR, $kg \cdot day^{-1}$).

$$TGR = (TFETWT - OLDWT) / OLDWT \qquad (14)$$

where OLDWT = previous day's fetus weight (kg).

Under some conditions (e.g., poor food quality or high activity costs) the net energy available to the cow for reproduction will be low. To account for these conditions, the model adjusts the target growth rate of the fetus according to the cow's energy balance and energy stores. It performs the adjustment following a series of decisions (i.e., Boolean logic operations). The first one is to determine if net energy available for reproduction (NETNRG) is greater than or equal to the energy required for growing the fetus at the target rate (EGEST). If this operation is true, the model will increase the fetus' weight (FETUSWT, kg) by an amount determined by the target growth rate (Eq. 15); any remaining energy is deposited as fat or lean tissue.

$$FETUSWT = FETUSWT + FETUSWT * TGR \qquad (15)$$

If net energy available for reproduction is less than the energy required, the model will determine if the energy deficit can be accounted for by the cow's energy reserves (i.e., fat stores).

The amount of energy that must be drawn from the cow's energy reserves to compensate for the energy deficit (kJ) is calculated as:

$$BFR = (EGEST - (NETNRG * 0.13)) / 0.84 \qquad (16)$$

where 0.13 is the efficiency of using fat for fetal growth (ARC 1980;88), and
0.84 is the efficiency of using body reserves for fetal growth.

The cow must have a critical amount of body fat before gestation can be maintained. That critical amount is expressed as:

CHECK5 = CHKWT * COWWT + (BFR / 39.54) / 1000 (17)

where: CHECK5 = critical amount of fat, below which female can not gestate,
CHKWT = the proportion of the cow's weight that must be maintained as a fat reserve (i.e., unusable fat reserve); determined by user (App. 1),
COWWT is the weight of the cow (kg),
39.54 = amount of kJ in 1 g of fat, and
1000 = converts results to kg.

The model compares CHECK5 with the amount of fat in the body reserves (FATWT, kg). If the amount of fat in the body reserves (FATWT) is less than the critical amount (CHECK5), the fetus is aborted (ABORT$ = "YES") and the pregnancy is terminated (PREG$ = "NO"). If it is greater than the critical amount, however, the model compares the amount of energy needed from the fat reserves for gestation (BFR) to the maximum amount of energy that can be taken from the reserves in a single day. That latter amount is calculated (but not stored) as,

68 * 39.54 * 0.84 (18)

where: 68 = maximum number of grams of fat that can be mobilized in a single day, derived from body weight trends of caribou on gross under nutrition. A drawdown of 68 g of fat is equivalent to 2,258 kJ,
39.54 = amount of kJ in 1 g of fat,
0.84 = efficiency at which caribou can use energy (ARC 1980:94)
from fat.

If the required energy (BFR) is less than the maximum amount available (Eq. (18)), the fetus will grow at the maximum rate (TGR) as calculated by Equation 15. If the required energy is greater than the maximum amount available, the model reduces the growth rate. This adjustment involves two steps. First, the model calculates the actual (as opposed to maximum defined in Eq. 18) energy available for gestation (NEWNET, kJ) from both fat stores and net energy intake (NETNRG):

NEWNET = NETNRG * 0.13 + (68 * 39.54) * 0.84 (19)

It then uses NEWNET to adjust the target growth rate and calculates the adjusted fetus weight as,

FETUSWT = (NEWNET / EGEST) * TGR * FETUSWT + FETUSWT (20)

If the actual energy available for gestation (NEWNET) is equal to the energetic cost of gestation (EGEST), the fetus grows at the target rate (i.e., 1 * TGR * FETUSWT). If the actual energy available is less than the energetic costs of lactation, then the fetus will grow at a reduced rate.

The model then calculates the actual (as opposed to target) energy cost of gestation by the product of CONCCOST (Eq. 13) and fetus weight.

$$EGEST = CONCCOST * FETUSWT \quad (21)$$

The weight of the conceptus (CONCWT, kg), which includes fetus weight and maternal fluids and tissue, is determined by,

$$CONCWT = FETUSWT * CWT \quad (22)$$

where: CWT = factor set by user (App. 1) that relates weight of fetus to weight of conceptus.

Pregnancy is completed when the number of days the cow is pregnant (DPREG) equals the gestation length (GESLEN) set by the user. At the iteration representing parturition, PREG$ is set to "NO" and the cow's weight is reduced by the weight of the conceptus.

$$COWWT = COWWT - CONCWT \quad (23)$$

Fetus weight is then assigned to calf weight (CALFWT) and the model begins the lactation portion of the simulation.

2) Lactation

If the female produces a viable calf (>3 kg) and the calf's age is less than the length of lactation (LACLEN, days; set by user), the model considers the female to be lactating. In a manner similar to that described for gestation, the model calculates targets for milk production based on the calf's age using equations provided by White and Allaye-Chan (pers. comm.).

For calves less than 10 days old:

$$TARMP = 0.131 * CAFAGE + 0.6 \quad (24)$$

For calves between 11 and 21 days old:

$$TARMP = 0.0291 * (CAFAGE - 10.) + 1.91 \quad (25)$$

For calves between 22 and 35 days old:

$$TARMP = -0.0586 * (CAFAGE - 21.) + 2.23 \quad (26)$$

For calves greater than 35 days old:

$$TARMP = -0.01241 * (CAFAGE - 35.) + 1.41 \quad (27)$$

where TARMP is target milk production ($ml \cdot day^{-1}$)
CAFAGE is the age of the calf (days).

The amount of energy contained in the milk is calculated as:

$$MLKNRG = 12.06 - 0.00324 * (TARMP * 1000.) \qquad (28)$$

where MLKNRG is the amount of milk energy (kJ)

As in the gestation section described earlier, the model adjusts target milk production during periods when the cow's energy balance can not meet target demands. To determine if adjustment is necessary, the model steps through a series of decisions similar to those described for gestation. The first decision is to determine if the energetic cost of target lactation (ELACT) can be met by the available energy (NETNRG). Costs of target lactation are determined from the energy contained in target milk production and the efficiency of producing that energy.

$$ELACT = MLKNRG * TARMP * 1000 \qquad (29)$$

Where ELACT is the energy contained in target milk production (kJ)

If there is enough available energy from MEI and fat reserves, then actual milk production (MP, $ml \cdot day^{-1}$) will equal the target production.

$$MP = TARMP \qquad (30)$$

If there is not enough available energy, then the model will calculate the amount of energy that must be drawn from the cow's fat reserves to compensate for the deficit as in Equation (16) (substituting ELACT for EGEST and substituting EFLACT for 0.13 as the efficiency of using dietary energy for lactation). The model then calculates whether the cow has enough fat reserves to meet the lactation demand using Equation (17). As with gestation, the model compares CHECK5 with the amount of fat in the body reserves (FATWT, kg). If the amount of fat in the body reserves (FATWT) is less than the critical amount (CHECK5), the cow produces milk only to the level permitted by dietary net energy and body reserves. That available energy is calculated as:

$$NEWNET = NETNRG * EFLACT + (BFA * 39.54) * 0.84 \qquad (31)$$

Where NETNRG is energy available from diet (kJ),
EFLACT is the efficiency of using body reserves for lactation (proportion),
BFA is body fat available for lactation (g) and is calculated as:

$$BFA = (FATWT - (CHKWT * COWWT)) * 1000)$$

Where FATWT is the fat reserves of the cow (kg)
CHKWT is the proportion of the cow's weight that can not be catabolized,
COWWT is the weight of the cow (kg),

If fat reserves are greater than the critical amount, CHECK5, the model compares the amount of energy needed from the fat reserves for lactation (BFR) to the maximum amount of energy that can be taken from the reserves in a single day (defined in Eq (18)). If the required energy (BFR) is less than the maximum amount available (Eq. (18)), the cow will produce milk at the target level (i.e., MP = TARMP). If the required energy is greater than the maximum amount available, the model reduces the cow's milk production. It accomplishes that task by first calculating the target rate of daily milk production (re-uses TGR) as:

TGR = (TARMP - OLDMP) / OLDMP (32)

where OLDMP = previous day's milk production ($ml \cdot day^{-1}$)

and then adjusts that target growth rate using equations (19) and (20) (substituting ELACT for EGEST, EFLACT for 0.13, and OLDMP for FETUSWT) to produce the actual milk production for that day:

MP = (NEWNET / ELACT) * TGR * OLDMP + OLDMP (33)

Once the model has computed the actual milk production it calculates the actual energy cost of lactation by the equation:

ELACT = MP * MLKNRG * 1000 (34)

Body fat reserves of the cow are then adjusted for the actual costs of lactation

Calf growth rates are calculated from the following age-specific equations relating growth (GR, $kg \cdot day^{-1}$) to milk production (White and Allay-Chan pers. comm.):

For calves 1 to 21 days old,

GR = (((MP* 1000.) -653.)/2.79)/1000 (35)

For calves between 22 and 42 days old,

GR = MP/3.13 (36)

For calves greater than 42 days old,

GR = MP/2 (37)

Calf body weight (kg) is calculated as:

CALFWT = CALFWT+GR (38)

D. Growth and Fattening

The model uses Fancy's (1986) approach to determine growth from net energy balance.

The model calculates the daily energy balance (EB, kJ) by adjusting the energy requirements for activity (EACTIVE), gestation (EGEST), and lactation

(ELACT) by their respective efficiencies (EFMAIN, EFGEST, and EFLACT). Those adjusted energetic costs are added to the cost of maintenance calculated earlier (Eq. 3) and the resulting total is then subtracted from MEI to produce the cow's daily energy balance (EB).

EB=MEI-(MAINCST+ACTCST+GESTCST+LACTCST) (39)

Where EB is energy balance ($kJ \cdot day^{-1}$),
MEI is metabolizable energy intake ($kJ \cdot day^{-1}$)
MAINCST is the cost of maintenance($kJ \cdot day^{-1}$)
GESTCST is the cost of gestation ($kJ \cdot day^{-1}$), and
LACTCST is the cost of lactation ($kJ \cdot day^{-1}$)

If energy balance is negative, (i.e, an energy deficit), then catabolism occurs and the fat and protein reserves are reduced. If not, then energy is added to those body reserves.

Because MEI is used for growth and fattening at a lower efficiency than it is for maintenance, the model reduces the energy available for growth and fattening (REQPROD, $kJ \cdot day^{-1}$) by using the equation:

REQPROD=EB * EFPROD) (40)

where EFPROD is the efficiency of using metabolizable energy for growth and fattening and is calculated from the metabolizability of the diet by the following equation:
EFPROD=(0.78*QM)+0.006 (ARC 1980).

The amount of energy for growth and fattening (REQPROD) is either added to or subtracted from the cow's protein and fat reserves depending on whether it has a positive or negative value. The method Fancy (1986) used for adjusting those body reserves was not based on caribou. He used data from Torbit et al. (1985) on differential losses of fat and protein by mule deer during winter and applied the results to caribou. His approach is probably valid because composition of weight gains and losses have been shown to be similar among most adult ruminants (ARC 1980).

Fancy (1986) assumed that 27% of the energy content of weight gains or losses in caribou is contributed by the deposition or catabolism of body protein (PCTPRT), whereas 73% of that energy is associated with fat reserves (PCTFAT). The energy content of fat and dry protein used in the model is 39.75 $kJ \cdot g^{-1}$ and 23.85 $kJ \cdot g^{-1}$, respectively. Using these values, the number of grams of protein (GPRTN) and fat (GFAT) added or subtracted from body reserves is calculated as:

GPRTN = PCTPRT * REQPRO / 23.85 (41)

While the number of grams of fat added to or subtracted from body reserves is calculated as:

GFAT = PCTFAT * REQPRO / 39.75 / 1000 (42)

To provide for realistic gains and losses in fat reserves, the model restricts the maximum amount of fat loss to 68 g a day (derived from body weight trends of caribou on gross under nutrition). The magnitude of changes in protein reserves is not limited.

The main site of protein deposition is in muscle, which contains 29% protein and approximately 71% water (Fancy 1986). The weight of lean tissue catabolized can therefore be computed as:

$$GLEAN = GPRTN / 0.29 / 1000 \quad (43)$$

The weight of fat reserves is adjusted each day by adding or subtracting GFAT (kg):

$$FATWT = FATWT + GFAT \quad (44)$$

Weight of lean tissue is similarly adjusted by the value GLEAN (kg):

$$LEANWT = LEANWT + GLEAN \quad (45)$$

The variable WATRWT (kg) monitors seasonal changes in the weight of the body water pool. Fancy (1986) assumed a water conversion mechanism where water replaced catabolized body fat and protein. When fat is catabolized WATRWT is calculated as:

$$WATRWT = WATRWT - (GFAT * CFAT) \quad (46)$$

where CFAT is the proportion of fat replaced by water.

Similarly, when lean tissue is catabolized, the amount of water added to the body water pool is computed as:

$$WATRWT = WATRWT - (GLEAN * CLEAN) \quad (47)$$

where CLEAN is the proportion of muscle replaced by water.

The total weight of the cow (COWWT) is the sum of the 4 body components: fat weight (FATWT, kg), lean weight (LEANWT, kg), weight of the conceptus deposited (CONCWT), and water weight (WATRWT, kg). From COWWT, metabolic weight (WT75) is calculated as:

$$WT75 = COWWT^{0.75} \quad (48)$$

The model then stores the above results in the plot variables (Z) and begins a new iteration.

WEAKNESSES OF THE MODEL

The GROWTH model has at least three major weaknesses:

1) The growth efficiencies are not linked to diet quality.

2) The efficiencies for building body reserves are the same as for depleting them.

3) The model does not yet consider effects of nitrogen balance on the cow's energy balance.

The efficiencies are related to QM (ARC 1980), the metabolizability coefficient, but presently QM is not related to gross energy intake. When the ENERGY and GROWTH models are linked, this problem should be corrected.

Intuitively, one would not expect that using energy to build something up is as efficient as using it to break something down (supported by Second Law of Thermodynamics). In the model, however, the two efficiencies are equal. We believe that this equality is not correct.

We have included nitrogen dynamics in another version of the GROWTH model, but that version still needs considerable work.

REFERENCES

Agricultural Research Council (ARC). 1980. The Nutrient Requirements of Ruminant Livestock. Commonwealth Agricultural Bureaux, Slough, England. 351 pp.

Fancy, S.G. 1986. Daily energy budgets of caribou: a simulation approach. Ph.D. Thesis. University of Alaska, Fairbanks, Alaska. 226 pp.

Hudson R.J. and R.J. Christopherson. 1985. Maintenance metabolism. pp 121-142 in R.J. Hudson and R.G. White (eds.), Bioenergetics of Wild Herbivores. Boca Raton, Florida. CRC press. 314 pp.

Robbins, C.T. 1981. Wildlife Feeding and Nutrition. Academic Press, New York. 343 pp.

Torbit, S.C., L.H. Carpenter, D.M. Swift, and A.W. Alldredge. 1985. Differential loss of fat and protein by mule deer during winter. J. Wildl. Manage. 49:80-85.

APPENDIX 1

Operation

The following is a description of the operation of the GROWTH model. Appendix 2 lists all the variables used in the model. Interactive graphics will not be discussed; that part of the program is detailed in PC_SIMULATOR (Hovey 1988). The GROWTH model requires one set file and one data file.

The set file contains the following thirty-three statements:

Statement	Description
A	Parameter controlling the period and, in part, the horizontal shift of the sine curve that predicts metabolizable energy intake (MEI).
B	Parameter controlling, in part, the horizontal shift of the sine curve that predicts MEI.
BIRWT	Target birth weight of caribou calf (kg).
CDATE	Julian date of conception.
CWT	Factor relating conceptus weight and fetus weight.
EIMIN	This parameter sets the lower peak of the sine curve predicting MEI.
EIMAX	This parameter sets the upper peak of the sine curve predicting MEI.
MNMEI	Maximum daily metabolizable energy intake ($kJ \cdot day^{-1}$).
CLEAN	Proportion of lean tissue that can be replaced by water.
CFAT	Proportion of fat that is replaced by water.
FATWT	Initial amount of fat caribou begins simulation with (kg).
LEANWT	Initial amount of lean tissue caribou begins simulation with (kg).
WATRWT	Initial amount of water weight caribou begins simulation with (kg).
CONCWT	Initial weight of the conceptus (kg).
COWWT	Initial cow weight (kg).
GESLEN	Length of gestation period (days).

LACLEN	Length of lactation period (days).
PCTFAT	Proportion of energy gain or losses that comes from fat.
PCTPRT	Proportion of energy gain or losses that comes from protein.
QM	Proportion of the gross energy of the diet that is metabolizable (metabolizability coefficient).
COST(FEED)	Energetic cost of feeding ($kJ \cdot kg^{-1} \cdot hour^{-1}$).
COST(STAND)	Energetic cost of standing ($kJ \cdot kg^{-1} \cdot hour^{-1}$).
COST(WALK)	Energetic cost of walking ($kJ \cdot kg^{-1} \cdot hour^{-1}$).
COST(RUNNIN)	Energetic cost of running ($kJ \cdot kg^{-1} \cdot hour^{-1}$).
COST(LIE)	Energetic cost of lying ($kJ \cdot kg^{-1} \cdot hour^{-1}$).
COST(PAWINT)	Energetic cost of pawing ($kJ \cdot kg^{-1} \cdot hour^{-1}$).
CHKWT	Proportion of fat reserves that can not be catabolized.
SDAY(1-7)	Mid-point day of the first 7 seasons.
SDAY(8-15)	Mid-point day of the last 8 seasons.
SNODEP(1-7)	Snow depths for the first 7 seasons.
SNODEP(8-15)	Snow depths for the last 8 seasons.
SDPROP	Proportion of the snow depth that the cow sinks.
TIMESTOP	Variable that determines the day of the simulation the model is to interrupted.

The above data file names are existing data file names. The user can give data files any valid DOS file names.

The GROWTH model also uses one of the following 6 data files on the activity budgets of caribou (see ENERGY model App. 1 [Tables 1 & 5] for a description of data organization):

Data File Name	Description
ACTHIHI.DAT	Activity budgets for high insect harassment and severe winters.
ACTHILO.DAT	Activity budgets for high insect harassment and mild winters.

ACTLOHI.DAT	Activity budgets for low insect harassment and severe winters.
ACTLOLO.DAT	Activity budgets for low insect harassment and mild winters.
HARASHI.DAT	Activity budgets for high, prolonged insect harassment and severe winter.
HARASLO.DAT	Activity budgets for high, prolonged insect harassment and mild winters.

Besides prompting for the names of the set and activity budget files described above, the GROWTH model will ask the user:

"Is the cow pregnant (Yes or <CR>)?"

If the user presses the return key in response to this question (i.e., a "NO" answer) the model will begin the simulation by treating the cow as non-pregnant. A "YES" answer will have the opposite effect.

Appendix 2

Variable List for GROWTH Model

Variable	Description
A	Parameter controlling the period and, in part, the horizontal shift of the sine curve that predicts metabolizable energy intake (MEI).
ABORT$	Logical variable that monitors whether fetus has been aborted (Yes or No).
ACTBUG(7,15)	Activity budget data for 7 activities and 15 seasons
ACTDIM	Parameter indicating number of activities.
ACTIVE(7)	Array storing the 7 activities
B	Parameter controlling, in part, the horizontal shift of the sine curve that predicts MEI.
BFR	Amount of body fat required to meet pregnancy or lactation costs (kg)
BIRWT	Target weight of caribou calf at birth (kg)
CAFAGE	Age of the calf (days)
CALFWT	Weight of the calf (kg)
CDATE	Julian date of conception.
CFAT	Proportion of fat that is replaced by water during catabolism.
CLEAN	Proportion of muscle that is replaced by water during catabolism.
CHECK5	Critical amount of body fat required before cow can gestate or lactate (kg).
CHKWT	Proportion of fat reserves that can not be catabolized.
CONCCOST	Cost of pregnancy (kJ·kg fetus weight)
CONCWT	Weight of the conceptus (kg)
COST(7)	Array storing energetic costs of the 6 activities. All costs are on $kJ \cdot kg^{-1} \cdot hour^{-1}$, except pawing cost, which is on a daily basis.
COSTDIM	Parameter indicating number of activity costs

COWWT	Weight of the cow (kg)
CSTNRG(7)	Array storing energetic costs on a daily basis for all activities
CWT	Factor to calculate conceptus weight from fetus weight
DAY	Number of days since the beginning of the simulation
DPREG	Number of days the cow has been pregnant
DSNODEP	Daily snow depth (cm) linearly interpolated from seasonal data (SNODEP())
EACTIVE	Combined energetic cost of all activities ($kJ \cdot day^{-1}$)
EATINT	Proportion of feeding time spent eating
EB	Daily energy balance (kJ)
EGEST	Daily energetic cost of gestation (kJ)
ELACT	Daily energetic cost of lactation (kJ)
EIMIN	Parameter for fitting sine curve to drive MEI
EIMAX	Parameter for fitting sine curve to drive MEI
EFGEST	Efficiency of using energy for gestation
EFLACT	Efficiency of using energy for lactation
EFMAIN	Efficiency of using energy for maintenance
FATWT	The amount of fat on the cow (kg)
FEED	Parameter that indicates cow is feeding.
FETUSWT	Weight of the fetus (kg)
GESLEN	Length of gestation period (days).
GESTCST	Energetic cost of gestation corrected for inefficiencies.
GFAT	Amount of fat growth added to or subtracted from FATWT (kg).
GLEAN	Amount of lean tissue growth added to or subtracted from LEANWT (kg).

GPRTN	Grams of protein (g) growth.
GR	Growth rate of the calf ($kg \cdot day^{-1}$)
HI	Heat increment ($kJ \cdot kg^{0.75}$)
HP	Heat production ($kJ \cdot kg^{0.75}$)
KEYCHECK$	PC_SIMULATOR variable
LACLEN	Length of lactation (days)
LACT$	Logical variable that monitors whether cow is lactating (Yes or No).
LEANWT	Weight of lean tissue (kg)
LIE	Parameter that indicates cow is lying.
MAINCST	Daily cost of maintenance (kJ)
MAXDIM1	Maximum number of plot variables used in PC_SIMULATOR
MAXDIM2	Maximum number of single iterations that the model can perform.
MEI	Daily metabolizable energy intake (kJ).
MILKFAT	Percentage of milk that is fat (%)
MILKNRG	Amount of energy in milk ($kJ \cdot l^{-1}$)
MILKPRO	Percentage of milk that is protein (%)
MNMEI	Maximum daily metabolizable energy intake ($kJ \cdot day^{-1}$).
MP	Amount of milk production ($ml \cdot day^{-1}$)
NETNRG	Net energy (kJ)
NEWNET	New net energy after adding energy from body fat reserves (kJ)
OLPMP	Previous day's (i.e., iteration) milk production ($ml \cdot day^{-1}$)
OLDWT	Calf weight on the previous day (kg).
PAWIN	Proportion of time cratering
PCTFAT	Proportion of energy gains or losses that comes from fat.

PCTPRT	Proportion of energy gains or losses that comes from protein.
PREG$	Logical variable that monitors whether the cow is pregnant (Yes or No).
PTOTAL	Total of all activity proportions. This variable is used to normalize activity budget.
QM	Metabolizability (proportion) of gross energy content of diet
RCAP	Capacity of the rumen (3600 g)
REQPRO	Energetic requirements of growth and fattening (kJ)
RUNNIN	Parameter that indicates cow is running.
SEASDIM	Parameter indicating number of seasons used in model.
SDAY(15)	The midpoint, in Julien Days, of each of the 15 seasons.
SDPRDP	Proportion of the snow depth that cow sinks.
SINKDEP	Cow's sinking depth in snow (cm).
SNODEP(15)	Snow depths during the 15 seasons (cm)
SNOWX	Factor to adjust for energetic costs of moving through snow
STAND	Parameter that indicates cow is standing.
TARMP	Target milk production (ml·day)
TIME	Number of days into the simulation
TIMESTOP	Variable that determines the day of the simulation the model is to interrupted.
TBIRWT	Target birth weight (kg)
TFETWT	Target fetus weight (kg)
TGR	target growth rate (proportion) of fetus or milk production.
WALK	Parameter that indicates cow is walking.
WATRWT	Weight of additional water put on in winter (kg)
WT75	Metabolic weight of the female caribou

Z	PC_SIMULATOR plot variables that store the results of the simulation
ZN$	Plot variable names

PC SIMULATOR

Version 2.0

by

Fred W. Hovey
(88.03.31)

Background

PC_Simulator (PC_SIM) is a program that makes simulation models easy to use. Among its many useful features is the ability to plot, display, or print the values of key variables at any stage of the simulation. The program was developed at U.B.C. by the author for use with BASIC language compilers. It has its origins with Dr. C.J. Walter's (U.B.C.; Zoology) MICROSIMCON program that was developed in the late seventies for use with Apple micro-computers and INTERPRETATIVE BASIC. In 1985, D. J. Vales (U.B.C.; Forestry) modified MICROSIMCON to make it compatible with IBM PCs. Among the features Vales added were an online help menu and the ability to display plot variable names and data. Those features have been retained and enhanced in PC_SIM.

MICROSIMCON was designed to run in INTERPRETATIVE BASIC; consequently, it is slow. To take advantage of the speed, compactness, and portability of programs generated by compilers I greatly modified MICROSIMCON's code. The program is now structured and modularized. In addition, many new features and error-handling routines that improve the use of simulation models on micros have been added. Those features and changes are detailed in Appendix I.

Operation

PC_SIM begins by asking the user how many iterations (e.g., years or days) the model is to cycle before entering graphics mode (Fig. 1). The user can input any value up to a maximum of 32,737 (2^{15}-1). The program then asks for the names of the data files needed to run the simulation model. In each case, the program provides a default file name that can be chosen simply by pressing the carriage return key, ENTER (indicated as <CR>).

The starting values of certain variables used in the simulation can be set via a special data file called the "set file". That file, like all other data files used by PC_SIM, is external to the program; hence any editor can be used to create or change it. Each line of the set file contains the name and initial value of the variable that is to be set by the program. For example,

caribou weight =,100

would initialize the variable representing caribou weight to a value of 100. In all cases, a comma must appear after, but not before the equals sign ("="). To set the initial values for variable arrays, the numbers must be separated by commas. For example,

caribou numbers (age 1 to 5) =,1000,800,600,500,300

YEAR = 0

Simulation begins at Year: 0
Finish Simulation at Year:? 100

What is the name of the set file (DEFAULT = HARVEST.SET)?

Figure 1. The initial PC_SIM screen. In this example the user has instructed the program to execute the model for 100 iterations (i.e., years).

sets the first element of an array containing the age-specific population of caribou to 1000 and the last element to 300.

Once the data and initial conditions are loaded, the program begins cycling through the model. You can interrupt the progress of the simulation at the end of any cycle or iteration, by pressing any key on the keyboard or by setting the variable "TIMESTOP" in the set file to the desired stop point. This feature is useful for monitoring the progress of the simulation and for changing any of the set or data files. When the program has been interrupted, it asks the user whether they wish to continue the simulation (Fig. 2). Any response (including <CR>) other than a character string beginning with "N"[1] or "n" signifies a yes answer (this command format is common throughout PC_SIM). A NO answer terminates execution of the program. If the user wishes to continue the simulation, the program will ask whether it should enter graphics mode to allow viewing the status of key variables. A NO answer indicates that the user wishes to change the data before continuing the simulation. If the latter answer is chosen the program will ask the user if they wish to change the data in the set file (Fig. 3). A "Y" or "y" will generate the prompt: "Do you wish to read another set file". If the answer is affirmative, the user has the option of changing the set data either by reading another file ("Y" or "y" answer) or by changing the data interactively (<CR>). Before continuing, the program will also ask the user whether to change the other data files used in the simulation.

After an interruption, program execution begins with the iteration following the one at which the model was stopped. To restart the program, the user must enter the number of iterations at which the simulation is to terminate or press enter to take the default value. By repeating the above procedures the user can interrupt the program, view model's output, or alter the data and set files as often as desired.

<u>Graphics</u>

When the program reaches the specified number of iterations, it enters graphics mode (Fig. 4) and allows the user to, among other things, plot the simulation variables. One of the more useful commands in this mode of PC_SIM is the "H" or "HELP" command that generates the help menu. The following is a more detailed description of that menu.

<u>COMMAND</u>	<u>DESCRIPTION</u>
1. <u>C</u>lear	clears the screen and removes any minimum or maximum scaling factors set with either a min or m# command.
2. <u>Q</u>uit	clears the screen and terminates program execution.
3. <u>H</u>elp	prints the help menu on the screen.

1/ Throughout PC_SIM command language, only the first letter of a command needs to be entered. That letter can be in either upper or lower case format.

YEAR = 5

Simulation begins at Year: 0
Finish Simulation at Year:? 100

PRESS ANY KEY TO INTERRUPT PROGRAM
Do you want to continue program execution (<CR> or No)?

Do you want to plot the current results (Yes or <CR>)?

Figure 2. The PC_SIM screen after the program has been interrupted. In this example the user has interrupted the simulation at year 5 and instructed the program to continue execution. The program now asks the user whether to plot the results of the simulation from years 1 through 5.

```
CURRENTLY YEAR = 5


DO YOU WANT TO START OVER (Yes OR <CR>)?
DO YOU WANT TO CHANGE THE VARIABLES IN THE SET FILE (Yes OR <CR>)? y
DO YOU WANT TO READ ANOTHER SET FILE (Yes OR <CR>)?



What is the vulnerability of yearling males (Default = 2)
```

Figure 3. Interactively altering the data during an interruption in program execution. This screen is generated if the user answers no to the last question shown in Fig. 2.

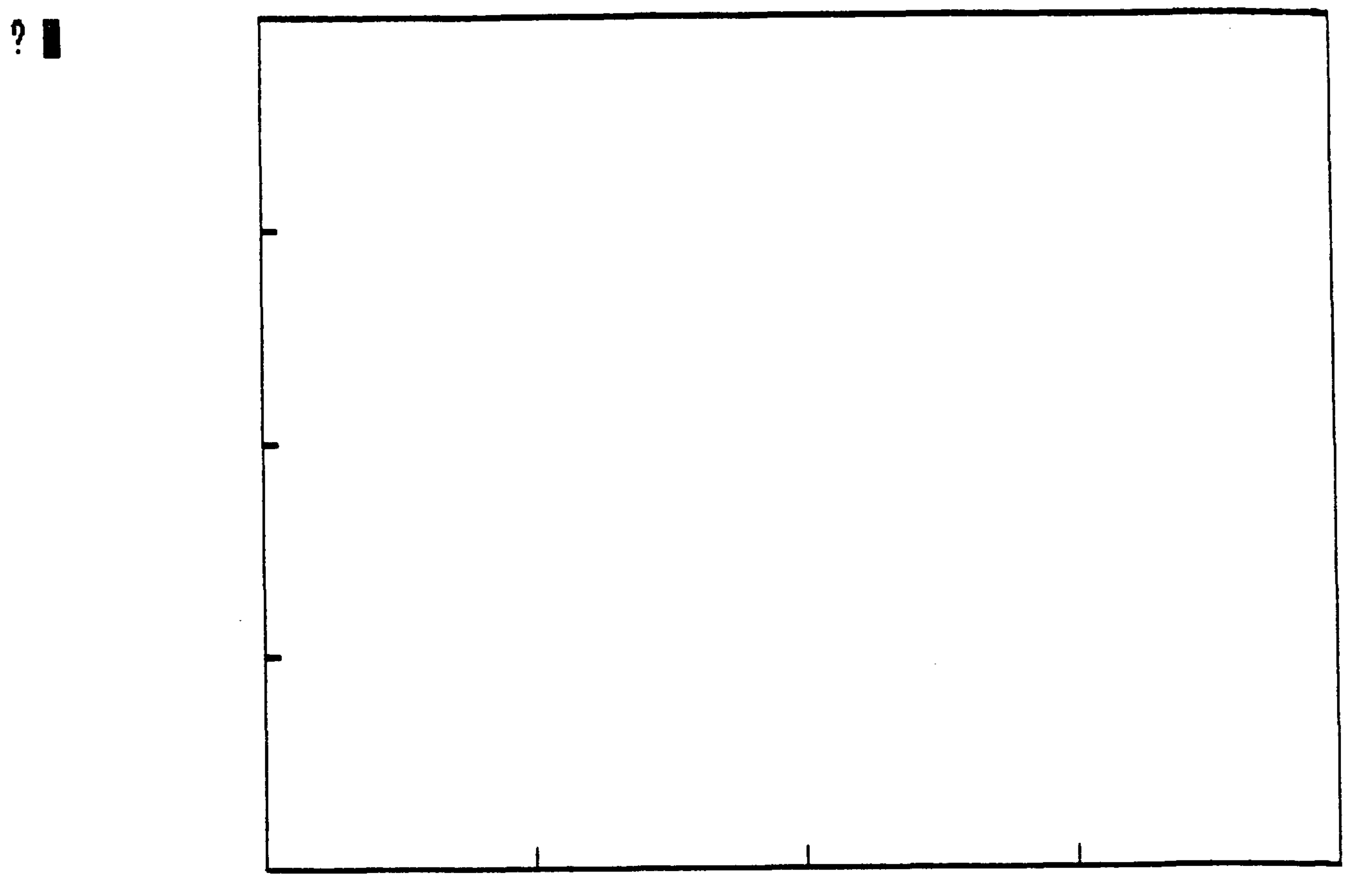

Figure 4. The PC_SIM screen after the program has entered graphics mode

4. Variable list — prints a list of all the variable names and their associated plotting numbers on to the screen.

5. Saves filename — saves the plot exactly as it appears on the screen onto a floppy or hard disk. The plot will be stored in the file specified by filename. That name must follow DOS conventions. If the three letter file extension is not specified, the program appends .BAS to the filename.

   e.g., S a:plotl saves screen onto a floppy disk located in drive A with the name plotl.bas.

6. Restore filename — erases the current screen and then restores the plot from hard memory that has been saved with an "S" command. The program allows additional plotting over the restored plot. With an "S" or "R" command, if the filename is not specified the program will trap the error and allow the user to enter DOS, quit the program, enter a new filename, or cancel the command. Experienced users of PC_SIM will take advantage of this feature when they wish to enter DOS without terminating program execution.

7. Display# — displays the values and associated iteration number for plot variable specified by #.

   e.g., D5 displays values for variable number 5

8. Maximum# — scales all subsequent plots to the maximum specified by #. This scaling factor applies only to those variables that have maximum values less than #.

   e.g., M1000 scales all subsequent plots to a maximum of 1000. This command stays in effect until a C command is issued.

9. MIN — scales all subsequent plots to the minimum value of the plot variable.

10. Write# filename	writes the values for variable # and their corresponding iteration numbers to the file specified by filename.
11. <cr> or Null line	restarts program for another simulation period.
12. #	plots variable # versus time e.g., 5 plots variable number 5
13. #p	plots variable # with unconnected points (scatter plot)
14. 1 vs 2 or 1 2	plots variable 1 versus variable 2 (i.e., 1 on Y axis, 2 on X axis)
15. 1/30 vs 2/100	plots variable 1 versus variable 2 with Y axis scaled to a maximum of 30 and the X axis scaled to a maximum of 100. Note that the "/" overrides any maximums set with the "M#" command (see point 8).
16. 1/30 vs 2p/100	same result as in the preceding example, except a scatter plot is generated instead of a line plot.

See Appendix I for a description of the other features of PC_SIM not included in the help menu.

Appendix I

Changes to MICROSIMCON:

1. The framework for reading data and set files has now been established. In addition to this new feature, for each required data set, the program offers the user the choice of taking a default file name (by simply pressing the return key) or specifying a different one.

2. Without terminating the simulation, the program now identifies and allows the user to correct such inevitable errors as:

    i) specifying an incorrect or non-existent file directory, file name, or DOS path.

    ii) specifying files in which the format of the data is incorrect.

    iii) writing plot or data files to protected disks or to disks with insufficient memory availability.

    iv) hardware or disk problems that prevent proper I/O operations.

3. On detecting an error or during plotting, the program now allows the user to access DOS without terminating simulation execution. Thus, the user now has the ability to edit incorrect data files, examine the file directories, change the default DOS path, etc., or issue any DOS command he wishes (this includes running other programs!) without re-starting the often time consuming simulation.

4. Simulations can now be interrupted at any point either by setting a special variable in the set file to a specified stop time or by pressing any key at the desired moment.

5. Interrupted simulations can be continued or terminated at the discretion of the user. If the user wishes to continue the simulation he can either enter graphics mode to plot the status of selected variables or change any of the data used in the simulation. Set data can be changed either interactively or by requesting a new set file.

6. The value determining the end of the simulation period can be changed before an interrupted simulation is continued.

7. The program now detects the kind of graphics card you are using automatically and then configures graphics arrays accordingly.

8. The labeling of all graphs has been made more readable; variable names, maximums, and minimums are now printed under all plotting situations. In addition, the bug where Y-axis labels containing exponents overwrote the screen during plotting has been corrected.

9. Negative numbers can now be plotted.

10. For variables containing both negative and positive values, the program now draws a "zero" line across the screen.

11. The program now allows the user to set a plot maximum so that all subsequent plots are on the same scale until either a clear command is entered or a new maximum is specified.

12. A bug has been corrected so that the user can now scale any variable independent of the rest to a maximum equal to or greater than the variable's actual maximum. This corrected feature also works when one variable is plotted against the other or when a plot maximum command has been issued.

13. The program now allows the user to set the minimum to which the program scales plots. By default all plots containing positive numbers are scaled from 0 to the maximum value of the data. When the "min" command is issued or when plots contain negative numbers, the program scales the axes from the smallest number to the largest or to the number set by "max" or "/" commands.

14. Values of selected variables indexed by corresponding day or year can now be displayed (100 at a time) for any simulation length (i.e., periods > 100).

15. The printing of variable names is no longer restricted to one screen.

16. More efficient programing structure and variable use has resulted in a saving of approximately 20% of the memory used in plotting.

17. A <u>single</u> simulation period can now extend to 32,767 (2^{15}-1) iterations.

18. After the user has finished plotting, the program now allows one the choice of either terminating the program or continuing the simulation. If the user decides to continue the simulation, he can either start over (i.e., re-initialize) or continue with the values the simulation terminated with. If the latter choice is exercised repeatedly, the user can theoretically simulate the chosen conditions for periods limited only by his patience. As with interrupted simulations, simulations following graphics mode also allow the user the option of changing any of the data or set files. If the user does not exercise the option of changing the end time, the simulation will, by default, finish after the same number of iterations as the previous simulation.