

FINAL REPORT

Effect of Climatic Variability on Freshwater Inflow, Benthic Communities, and Secondary Production in Texas Lagoonal Estuaries

A Final Report for Texas Water Development Board
Project 08-483-0791

By:

Hae-Cheol Kim, Ph.D.
and
Paul A. Montagna, Ph.D.



Texas A&M University-Corpus Christi
Harte Research Institute for Gulf of Mexico Studies
6300 Ocean Drive, Unit 5869
Corpus Christi, Texas 78412

Draft: September 2009
Final: May 2010

Cite as: Kim, H.-C. and P.A. Montagna. 2010. *Effect of Climatic Variability on Freshwater Inflow, Benthic Communities, and Secondary Production in Texas Lagoonal Estuaries. Final report to the Texas Water Development Board, contract number 08-483-0791. Harte Research Institute for Gulf of Mexico Studies, Texas A&M University-Corpus Christi, Corpus Christi, Texas. 119 pages.*

TABLE OF CONTENTS

Abstract.....	ii
1. Introduction	1
2. Methods	3
2.1 Databases.....	3
2.1.1. Study area.....	3
2.1.2. Benthic macrofauna	5
2.1.3. Water quality	5
2.1.4. Predators	7
2.1.5. Other environmental variables	7
2.2. Model description	8
2.3. Model equations.....	11
2.3.1. Governing equation for state variable	11
2.3.2. Dimensionless scaling factor of benthic growth (E_{ben}).....	12
2.3.3. Predation	18
2.4. Model calibration	19
2.5. Model validation	21
3. Results.....	22
3.1. Salinity and temperature differences among estuaries	22
3.2. Model validation for the time series (1988 – 2008).....	23
3.3. Model simulation results	26
3.3.1. Lavaca-Colorado Estuary (Lavaca and Matagorda Bay)	26
3.3.2. Guadalupe Estuary (upper and lower San Antonio Bay)	30
3.3.3. Nueces Estuary (Nueces and Corpus Christi Bay)	33
3.3.4. Laguna Madre Estuary (Baffin Bay and upper Laguna Madre)	36
4. Discussion	39
4.1. Ecosystem modeling and goodness of fit	39
4.2. Simulated benthic biomass.....	41
4.3. Salinity/nutrients change scenario and benthic responses	42
4.4. Lessons learned	51
5. Concluding Remarks	54
6. References	55
7. FORTRAN Programs.....	60
7.1 TEXAS.FOR.....	61
7.2 DEPOSIT.FOR	63
7.3 EPIGROW.FOR	68
7.4 XSIM0.FOR.....	73
7.5 XFORC.FOR.....	74
7.6 XTART.FOR	85
7.7 XCAL.FOR	87
7.8 XSENS.FOR.....	107
8. TWDB Review Comments	116

Effect of Climatic Variability on Freshwater Inflow, Benthic Communities, and Secondary Production in Texas Lagoonal Estuaries: FY2007-FY2008

Abstract

This is the final report for a 2-year project (TWDB contract numbers 07-483-0672 and 08-483-0791) with the objective to complete implementation of an ecological model to predict system-wide secondary production for two trophic groups of benthic organisms in response to different freshwater inflow regimes that result from the climatic ecotone along the Texas coast. The bioenergetic model was calibrated using an 11-year dataset (from 1988 to 1999) and validated with a 20-year data from (1988 to 2008) from four estuaries in South Texas: Lavaca-Colorado (LC), Guadalupe (GE), Nueces (NE) and Laguna Madre Estuaries (LM). The estuaries lie in a climatic gradient where LC and GE receive more rainfall than NE, and NE receives more rainfall than LM. Consequently inflow decreases along the gradient and salinity increases. In addition there is year-to-year variation in rain and inflow that results in wet and dry years. Therefore, this combination of the climatic gradient and temporal variability can be used to identify the effects of inflow on estuarine productivity.

Among Texas estuaries, increased salinity (and thus decreased inflow) benefited deposit feeders, while suspension feeders were harmed, but the net overall effect is a decrease in functional diversity when salinity is increased. Within estuaries, the benthic community of secondary bays is harmed by reduced inflow, whereas, the community in primary bays appears to benefit in biomass increase by reduced inflow. This is because lower salinity regimes are required to support food production for suspension feeders, and there are more polyhaline deposit feeding species. Freshwater inflow is important in to maintain productivity and functional diversity; and required to maintain functional, healthy estuarine ecosystems.

1. Introduction

The ecology of estuaries is strongly influenced by the quantity, timing, frequency, and duration of freshwater pulses to coastal ecosystems (Montagna et al. 2002a). In Texas, there is a strong climatic gradient with decreasing precipitation, and concomitant freshwater inflow, from northeast to southwest (Montagna et al. 2007). Along this gradient, rainfall decreases by a factor of two, but inflow balance decreases by almost two orders of magnitude. Inflow balance is the sum of freshwater inputs (gaged, modeled runoff, direct precipitation, plus return flows) minus the outputs (diversions and evaporation). The net effect is a gradient with estuaries with similar physical characteristics but a declining salinity gradient (Montagna et al. 2009).

Another characteristic of Texas estuaries is the extreme year-to-year variability of precipitation, inflow, and salinity, which is caused by the El Niño Southern Oscillation (ENSO) (Tolan 2007). The ENSO climate signals are correlated to salinity structure within Texas estuaries within 4 to 6 months. During El Niño events, salinities in Texas estuaries decrease because of increased freshwater flows to the coasts. During La Niña periods, salinities increase because of the drier climatic conditions. These cycles occur with a periodicity of 3.55, 5.33, and 10.67 years. The ENSO is dominated by the 3.55- and 5.33-year periods and the 10.67-year period is defined by the Pacific Decadal Oscillation. The combination of latitudinal and long-term climate differences (which drive inflow variability) and the varying geography and tidal dynamics of the estuaries is responsible for the uniqueness, or estuarine signature, common in estuaries throughout the world. Thus, the differences in inflow regimes among the Texas estuaries are driven by the combination of spatial and temporal climatic regime shifts.

The climate regime differences in Texas estuaries drives freshwater inflow (which is driven by precipitation and runoff) and biological processes also vary

along the same gradient. Thus, the objective of the current project is to link long term data bases on temperature, salinity, and macrofauna communities to predict how changes in climatic variability influence the structure and function of estuarine communities. The approach is to study benthos and inflow dynamics over a large regional scale for the long term to capture spatial and long term effects. Benthos are good indicators of environmental change, because they are fixed spatially, have relatively long life cycles, and are at the bottom of the food chain, thus integrating long term changes in the overlying water column (Montagna and Kalke 1995).

Benthic data has been collected in the Texas central coast estuaries from 1984 to present (as this is being written in 2009). At each quarterly sampling period, nutrient concentrations, primary producer biomass, and benthos biomass has been measured. The combination of benthos and water quality data provides the opportunity to investigate the role of freshwater inflow in driving ecosystem dynamics using a model framework that was first developed in 1995 (Montagna and Li 1996). The ecological model is mechanistic and calculates bay-wide productivity for two different trophic groups of benthos: those relying on primary producers in the water column for food, e.g., filter, interface, and suspension feeders; and those relying on organic matter in sediments for food, e.g., the deposit feeders and omnivores. The model framework was modified by Kim and Montagna (2009) to better resolve predator-prey interactions and responses to inflow. In the current project, the model has been refined again and is recalibrated using an 11-year dataset from 1988 to 1999 and validated with long-term data from 1988 to 2008. The project has been completed over a 2-year period from FY 2007 (TWDB contract numbers 07-483-0672) and FY 2008 (08-483-0791). The main activity in FY 2007 was data base assembly, and the current report is based on modeling activities accomplished during FY 2008.

2. Methods

2.1 Databases

2.1.1. Study area

Four estuaries were studied that are located in South Texas: the Lavaca-Colorado (LC), Guadalupe (GE), Nueces (NE) and Laguna Madre (LM) estuaries (Table 1). The sampling design is along salinity gradients among and within the estuaries. Within each estuary, two bays were sampled: the primary bay closer to the connection with the Gulf of Mexico, and the secondary Bay closer to the river source of the estuary (Fig. 1). There were at least two stations sampled in each bay (Table 1).

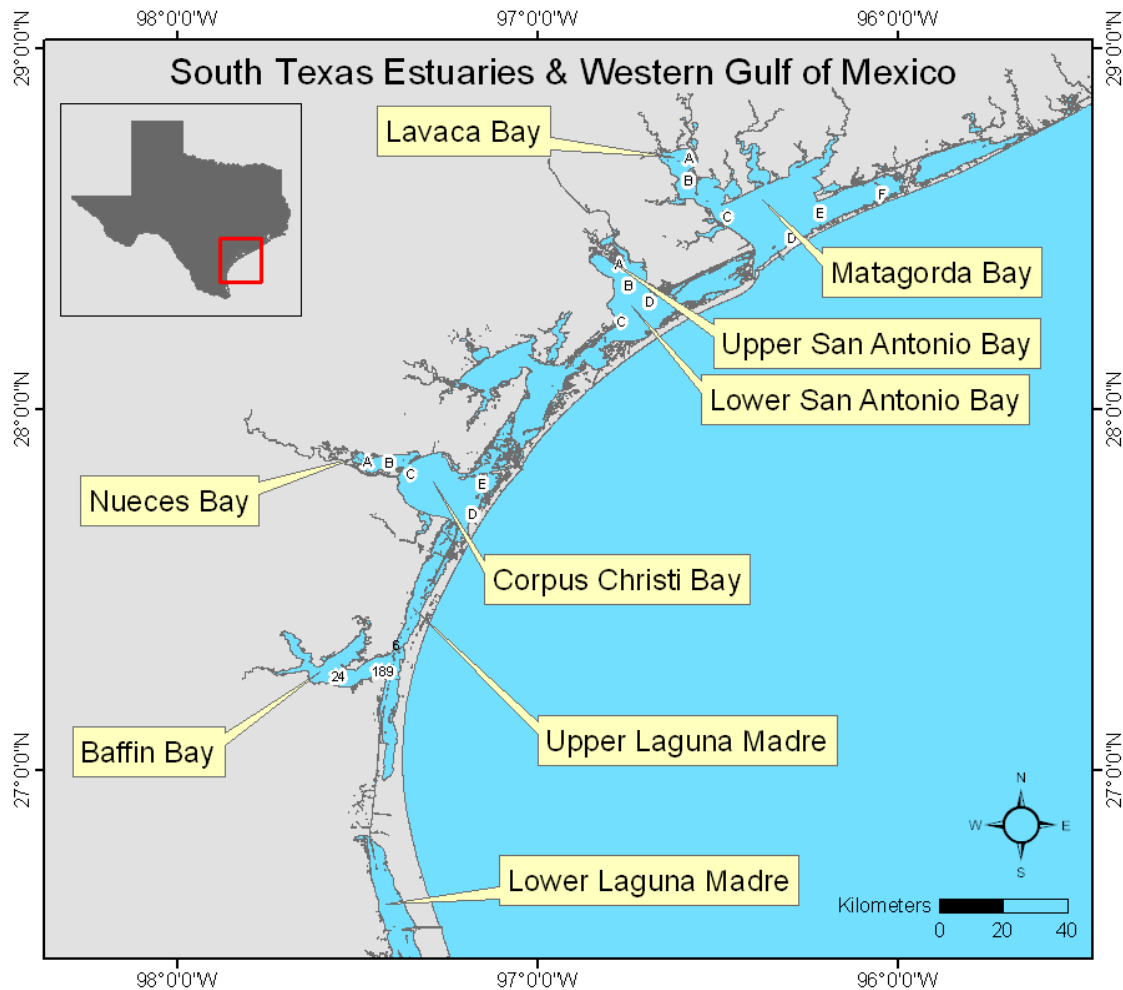


Fig. 1. Locations of long-term sampling stations in south central Texas bays.

Table 1. Sampling stations, locations and periods for the continuous long-term monitoring database in the eight Texas bays. Environmental and biological variables (temperature, salinity, nutrients, predator density and benthos biomass) used in the model were selected from the same database.

Estuary	Bay Name	Stations	Location	Period of Sampling	Period for Data Analyses (^a years)
Lavaca-Colorado Estuary	Lavaca	A	28° 40' 12" N, 96° 34' 48" W	1984 – 2009	1988 – 2008 (21)
		B	28° 38' 24" N, 96° 34' 48" W	1988 – 2009	1988 – 2008 (21)
	Matagorda	C	28° 32' 24" N, 96° 28' 12" W	1988 – 2009	1988 – 2008 (21)
		D	28° 28' 48" N, 96° 17' 24" W	1988 – 2009	1988 – 2008 (21)
		E	28° 33' 0" N, 96° 12' 36" W	1993 – 1995 2004 – 2007	1993 – 1995 (3) 2004 – 2007 (4)
		F	28° 36' 0" N, 96° 02' 24" W	1993 – 1995 2004 – 2007	1993 – 1995 (3) 2004 – 2007 (4)
Guadalupe Estuary	Upper San Antonio	A	28° 23' 37" N, 96° 46' 20" W	1987 – 2009	1989 – 2000 (12)
		B	28° 20' 52" N, 96° 44' 45" W	1987 – 2009	1989 – 2000 (12)
	Lower San Antonio	C	28° 14' 46" N, 6° 45' 54" W	1987 – 2009	1989 – 2000 (12)
		D	28° 18' 8" N, 96° 41' 4" W	1987 – 2009	1989 – 2000 (12)
Nueces Estuary	Nueces	A	27° 51' 39" N, 97° 28' 25" W	1987 – 2009	1988 – 2002 (15)
		B	27° 51' 26" N, 7° 24' 37" W	1987 – 2009	1988 – 2002 (15)
	Corpus Christi	C	27° 49' 31" N, 97° 21' 8" W	1987 – 2009	1988 – 2002 (15)
		D	27° 42' 46" N, 97° 10' 43" W	1987 – 2009	1988 – 2002 (15)
		E	27° 47' 50" N, 97° 9' 3" W	1987 – 2009	1988 – 2002 (15)
Laguna Madre Estuary	Baffin	6	27° 16' 37" N, 97° 25' 37" W	1988 – 2000	1988 – 2000 (13)
		24	27° 15' 50" N, 97° 33' 5" W	1988 – 2000	1988 – 2000 (13)
	Laguna Madre	189G 189S	27° 20' 60" N, 97° 23' 33" W	1988 – 2000	1988 – 2000 (13)

^anumber of years sampled

2.1.2. Benthic macrofauna

Long-term macrobenthos data used to perform the modeling experiment to determine the effects of alterations in freshwater inflow in these estuaries has been collected since 1987. From previous studies, it was learned that long-term changes in benthos within these estuaries could be characterized by sampling on a quarterly basis (Kalke and Montagna 1991, Montagna and Kalke 1992). A long-term (12 to 21 years) data set was pooled from each estuary for the comparison of simulation results (Table 1). During each sampling event, three independent macrobenthos samples were collected using 6.7-cm diameter sediment cores (35.4 cm² area) to a depth of 10 cm and preserved with 5% buffered formalin. The samples were sorted and dry weight biomass was measured by major taxa (see Kalke and Montagna 1991, Montagna and Kalke 1992 for details).

2.1.3. Water quality

Concurrent with the benthic samples, hydrographic measurements were also made during each sampling period, and these measurements included: chlorophyll *a*, nutrient concentrations, dissolved oxygen, salinity, temperature, and water depth.

Nutrient concentration for dissolved inorganic nitrogen (DIN; sum of nitrate, nitrite and ammonium), dissolved orthophosphate (PO₄³⁻), and silicate (SiO₄), is inversely related with salinity; and salinity is inversely related with inflow (Fig. 2 and slope *a* in Table 2). The inverse relationship between nutrients and salinity is driven by freshwater inflow changes.

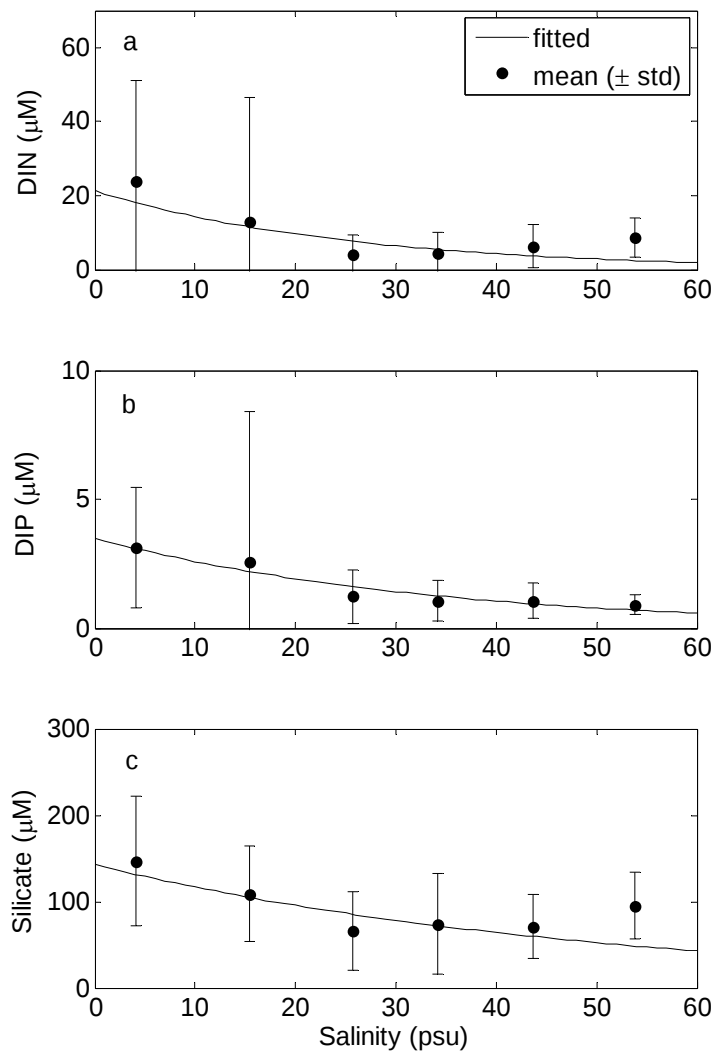


Fig. 2. Relationship between salinity and nutrients in the Lavaca-Colorado Estuary. (a) DIN (sum of nitrate, nitrite and ammonium), (b) orthophosphate (PO_4^{3-}) and (c) silicate (SiO_4). Salinity was binned every 10 psu and mean nutrient concentration (filled circles) with one standard deviation (error bar). Slope, Y-intercept and correlation coefficients are listed in Table 2.

Table 2. The results of linear regression among variables in the Lavaca-Colorado Estuary. *X* and *Y* represent independent and dependent variable, respectively. Slope (*a*), intercept (*b*), coefficient of determination (*r*²), and probability values (*p*) are listed. Abbreviations: DIN=dissolved inorganic nitrogen, DIP=dissolved inorganic phosphorus, Si=silicate.

<i>X</i>	<i>Log(Y)</i>	<i>a</i>	<i>b</i>	<i>r</i> ²	<i>p</i>
Salinity	DIN (μM)	-0.04	3.06	0.65	0.10
Salinity	DIP (μM)	-0.03	1.25	0.87	0.02
Salinity	Si (μM)	-0.02	4.96	0.76	0.05

2.1.4. Predators

Fisheries data from 1988 to 2009 were obtained from Texas Parks and Wildlife Department (TPWD). The TPWD, Coastal Fisheries Division samples monthly in the four estuaries (LC, GE, NE, and LM) using an otter trawl and gills nets. A study of stable isotopes and mercury bioaccumulation in different food chains determined that black drum, red drum, and blue crab are the main predators on benthic infauna (Montagna, unpublished data). Therefore, the average value for catch per unit effort (CPUE) of each of these three main predators was used. The crab data comes from the trawl samples and the fish data comes from the gill net samples.

2.1.5. Other environmental variables

Primary production was used as input to the model. However, there is very little primary production data from the study areas and no time series data at all. Therefore, primary production data was created in a model as a function of limiting factors (day length, temperature and nutrient concentration) from the historical maximum value. Previous studies report a range from 0.5 to 5 g C m⁻² d⁻¹ in the estuaries along the Texas coast (Armstrong 1985, Stockwell 1989, Brock 1994). To derive a forcing function that can simulate day length, a dataset of monthly day length for the Texas coastal area was obtained from Tony Amos, University of Texas Marine Science Institute.

2.2. Model description

The long-term benthic macrofaunal data sets from the four estuaries were used to calibrate the model of biological processes. The two principle environmental factors associated with freshwater inflow are salinity and nutrient concentrations; therefore, the relationship between biomass of benthic macrofauna and these environmental factors was incorporated into the model. To test for inflow effects, the ideal input to the model would be freshwater inflow as a forcing function driving a physical model of salinity. However, inflow rates have variable effects on salinity depending on the hydrological, hydrographic, and tidal characteristics of the estuary. Therefore, a physical model (i.e., a box model or hydrodynamic model) that predicts salinity change under varying inflow scenarios would be needed to provide input to the biological model. To avoid this level of complexity, the empirical salinity values were used as input, thus salinity was used as a surrogate for freshwater inflow. Salinity values represent the integration of all the physical characteristics of the estuary (e.g., size, inflow, outflow, residence time, tidal exchange, and climatic variability). Other input data to the model included fish and crabs as predators, temperature, water depth, day length, and nutrient concentrations.

Using energy circuit language (Odum 1971, 1983), a schematic of stores and flows in the benthos was conceptualized to guide model development (Fig. 3). There are two main trophic guilds in benthic sediments: the grazing food-chain and the detrital food chain (Tenore et al. 2006). Grazers consume autotrophic production and detritivores utilize heterotrophic production. To simplify the model, all macrobenthic animals were separated into one of two groups: the grazers, (hereafter called suspension feeders) to represent the grazing food chain, and the detritivores (hereafter called deposit feeders) to represent detritivores and omnivores. In Texas Lagoons, most of dominant taxa and ecologically important species fall into these two functional groups,

suspension and deposit feeders (see Table 3 in Tenore et al. 2006), and these two trophic groups are state variables in this modeling exercise.

Suspension feeders are those organisms obtaining their food by capturing suspended particles from the sediment surface or water column, filtering phytoplankton from the water column, or grazing benthic diatoms on the sediment surface. Suspension feeding taxa include the Mollusca, Crustacea, and chironomid larvae. Suspension feeders are members of the grazing food chain and thus are linked primarily with food produced by autotrophy.

Deposit feeders are defined as those organisms that obtain their food through ingestion of the sediment, predation, or omnivory. The deposit feeders include the Hemicordata, Nemertinea, Ophiuroidea, Polychaeta, and Sipunculida. Deposit feeders are members of the detrital food chain and are thus linked primarily with food produced by heterotrophy.

The simplification of classifying all organisms into one of two function groups allows suspension feeders to be defined as organisms limited by autotrophic food sources, and deposit feeders as organisms limited by heterotrophic food sources. This is important for setting up the model so that the drivers of benthic productivity are clearly delineated (Fig. 3).

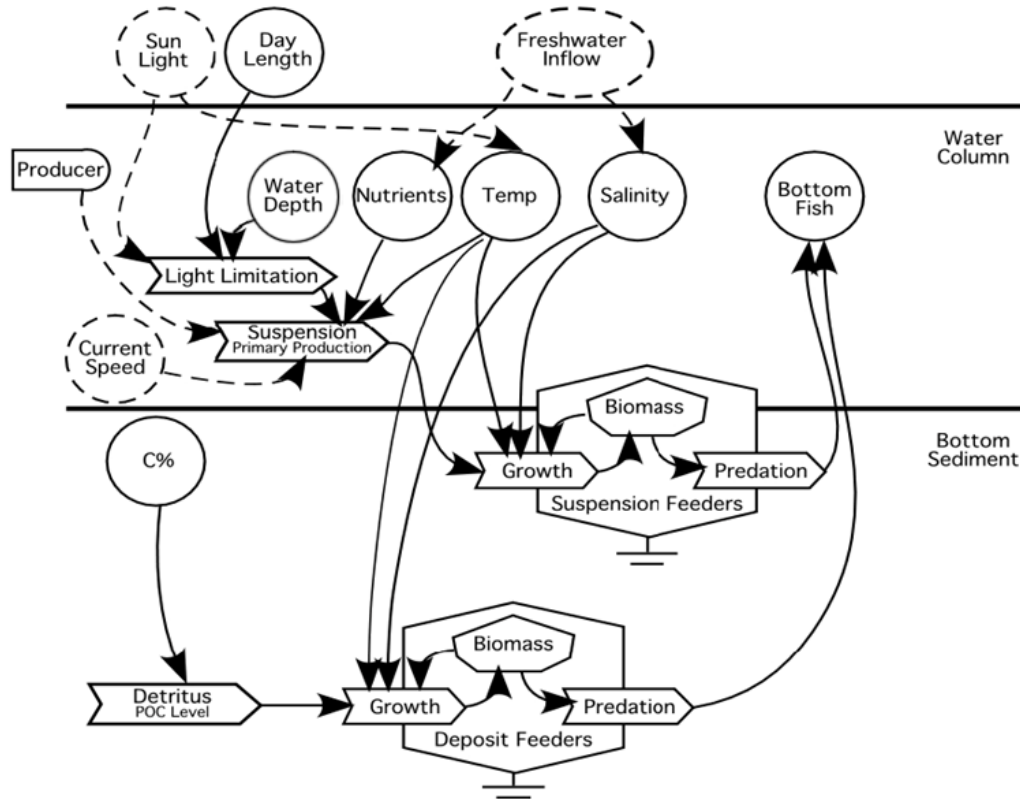


Fig. 3. Energy circuit diagram for the structure of the benthic macrofauna biomass model. Dashed lines represent variable not included in the model.

The model schematic (Fig. 3) also shows two major environmental (i.e., salinity and temperature) and two major biological/ecological factors (i.e., food sources and predators), which regulate both gain and loss of the biomass in the two state variables. Effects of these environmental factors were parameterized in terms of scaling factors representing environmental limitations. Other environmental variables (nutrients and day length) were used for estimating food source availability (i.e., primary production). Primary producers, whose growth is based on irradiance, temperature, and nutrient concentrations, are the main food source for suspension feeders. Deposit feeders primarily consume particulate organic matter (POM), and this can be approximated by the concentration of total organic carbon (TOC) in sediments, which is based on data (Montagna, unpublished, Montagna 2000). However, in the present study, the

dynamics of POM, nutrients, and salinity caused by mixing and benthic-pelagic feedbacks were not simulated. Instead, the measured POM, concentration of nutrients, and salinity were directly used as input data, assuming that the observations of these properties are reflection of physical and biogeochemical mixing features in the estuarine system.

2.3. Model equations

2.3.1. Governing equation for state variable

The governing equation uses a template of Lotka-Volterra growth model (Lotka, 1925) that models a density-dependent logistic growth of a population (Brown and Rothery, 1993):

$$\frac{dB}{dt} = r \cdot B \cdot \left(1 - \frac{B}{c}\right) - g \cdot F \quad (1)$$

where r is the maximum net growth rate of benthos (B) without predation pressure. c is the biomass carrying capacity for a population that is limited by space. The predation loss is calculated by the feeding rate of predators, g , and the density of predatory crab and fish, F . Therefore, Eq. (1) has a unit in biomass (mg or g) over time (day or month). In general, the net growth rate implicitly represents the filtering rate, ingestion rate, assimilation efficiency, respiration rate, aging mortality, and excretion rate; and the only loss of benthos biomass is by the predation rate.

However, growth rates in a population are also influenced by many other environmental effects. In this study Eq. (1) was modified to include environmental limitations as dimensionless scaling factors (E 's), which range between 0 and 1. When $E = 1$, there is no environmental limitation, and the benthic population can obtain a maximal growth rate or the predators can reach a maximal feeding rate. When $E = 0$, environmental factors reach maximum limitation, benthic populations do not grow, or predators do not consume

benthos. The governing equation of the model is as follows:

$$\frac{dB_{(i,j)}}{dt} = r_{(i)} \cdot E_{ben(i,j)} \cdot B_{(i,j)} \cdot \left(1 - \frac{B_{(i,j)}}{c_{(i)}}\right) - g_{(i,j)} \cdot \sum_k F_{(k,j)} \quad (2)$$

where $i = 1 - 2$ for deposit feeders or suspension feeders, $j = 1 - 2$ for the two bay systems (primary and secondary bay), $k = 1 - 3$ for three different predators (red drum, black drum and blue crab), $r_{(i)}$ is the monthly net growth rate (month⁻¹), $E_{ben(i,j)}$ is the environmental limitation for benthic biomass growth (dimensionless), $c_{(i)}$ is the biomass carrying capacity levels for the two feeding groups (mg dw m⁻²), and $g_{(i,j)}$ is the temperature- and density-dependent predation rate (g dw m⁻² month⁻¹ individual⁻¹) by fish k in bay j to prey benthos i . $F_{(k,j)}$ is average density for predator fish k (number of individuals).

The next two sections describe the parameterizations of rate processes (e.g., growth and predation) and effects of limitations (e.g., temperature, salinity and density) in the governing equation (Eq. 2), and these parameters will be derived for deposit and suspension feeders in all four bays. For convenience, species and location index, i and j , will not be included in the parameterizations described in the following sections.

2.3.2. Dimensionless scaling factor of benthic growth (E_{ben})

Benthic growth can be controlled by three environmental variables: temperature, salinity and food availability. Thus, the term E_{ben} in Eq. (2) can be parameterized with respect to temperature (E_{tem}), salinity (E_{sal}), and food concentration limitation (E_{food}):

$$E_{ben} = E_{tem} \cdot E_{sal} \cdot E_{food} \quad (3)$$

An Arrhenius-type exponential equation was used to represent the limiting effect of temperature (Carrada, 1983). This formula can provide an accurate temperature-dependent metabolism function and is used in estimating temperature-dependent growth rate:

$$E_{tem} = \frac{1}{e^{\frac{|T-T_{1opt}|}{T_{w_ben}}}} \quad (4)$$

where E_{tem} is the temperature limitation, T is the temperature (°C), and T_{1opt} is the most suitable temperature (°C). When $|T - T_{1opt}|$ is close to T_{w_ben} , $E_{tem}=1$, and there is no temperature limitation. Therefore, T_{w_ben} is a parameter (°C) that describes the weighting due to temperature limitation, and higher T_{w_ben} leads to higher sensitivity of E_{tem} to temperature (Fig. 4).

Salinity is one of the most influential environmental variables affecting benthic communities and is directly correlated with FWI. All invertebrates have optimal salinity ranges at which population growth is maximal (Wohlschlag et al., 1977). We used an Arrhenius exponential function in the model in order to represent salinity limitation:

$$E_{sal} = \frac{1}{e^{\frac{|S-S_{opt}|}{S_{w_ben}}}} \quad (5)$$

where E_{sal} is the salinity limitation, S is salinity (psu), S_{opt} is the optimal salinity (psu) for a population, and S_{w_ben} is a parameter (psu) that describes the weight of the salinity limitation. There is no salinity effect when $S_{w_ben}=\infty$. Salinity limitation has a centralized optimum, with greater effects at high and low salinities. The greater the salinity tolerance range is, the higher is the S_{w_ben} value (Fig. 4). The equation has the same form as that used for temperature

limitation, but the parameters (T_{opt} , S_{opt} , T_{w_ben} , S_{w_ben}) were calibrated independently.

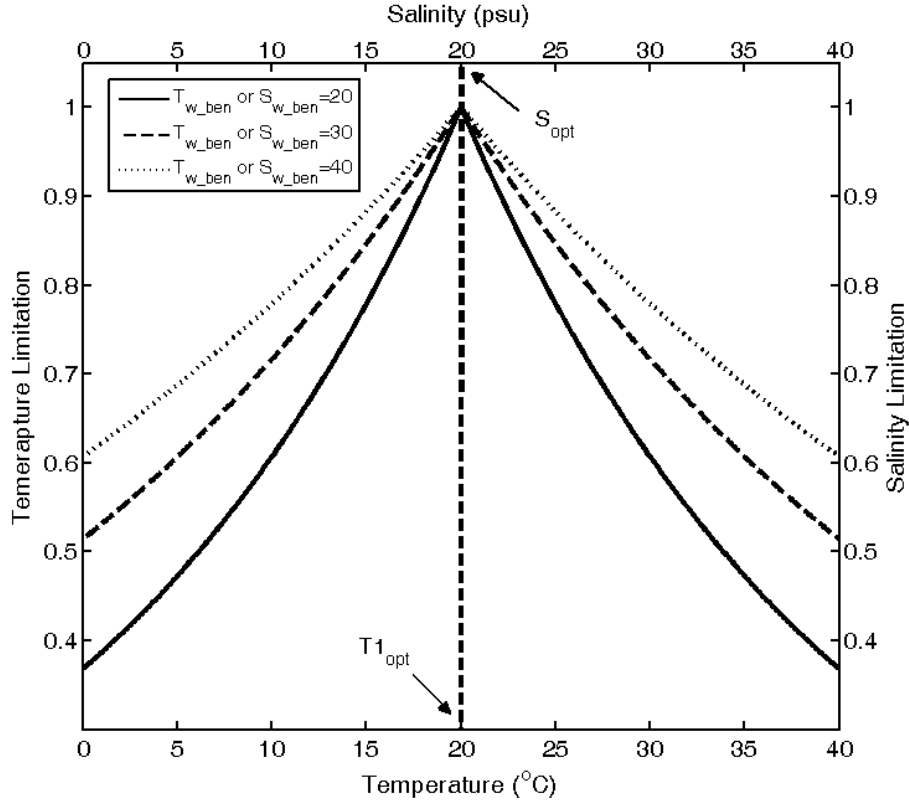


Fig. 4. Limitation by temperature and salinity. Optimal temperature (T_{opt}), salinity (S_{opt}) and sensitivity parameters (i.e., T_{w_ben} , S_{w_ben}) are based on Eqs. (4) and (5), respectively, and these parameters were calibrated independently.

The limitation of food source was described with a Michaelis-Menten type of uptake kinetics (Keen and Spain, 1992):

$$E_{food} = \frac{M_{(i)}}{M_{(i)} + K_M} \quad (6)$$

where E_{food} is the food limitation, $M_{(i)}$ is the concentration of the food source for benthic organisms ($i = 1, 2$), and K_M is a parameter at which the food concentration is at half the maximum level of the population growth rate.

As two feeding groups are simulated in the model (deposit and suspension feeders), there are two different food sources: detritus in sediment and organic matter in the water column. Sedimentary POM was used as a food source for deposit feeders, and expected primary production was used for suspension feeders.

The POM levels (p_{pom}) were calculated from the measured percent carbon content in the sediment samples (C%) for two bays:

$$C(\%) = \frac{p_{poc}}{p_{sed}} \cdot 100 \quad (7)$$

where p_{poc} is the sedimentary POC level (g C m⁻²) for each bay, and p_{sed} is an average dry weight of the whole sediment (g dw) per core sample (where area and depth of the core is 35.4 cm² and 10 cm, respectively), which was set at 16.4 (Montagna, pers. comm.). The measured mean C% from CHN analyzer and the calculated p_{pom} ($p_{poc} \div 0.42$) are listed in Table 3 for each bay. These sedimentary POM levels represent the food sources available for deposit-feeders ($M_{(1)}$) in each bay:

$$M_{(1)} = p_{pom} \quad (8)$$

Primary production is expected to be the most important food source for suspension feeders ($M_{(2)}$). Primary production is simulated as a function of day length, temperature, and nutrient concentration:

$$M_{(2)} = F \cdot p_{pp} \cdot \frac{1}{e^{\frac{T - T_{opt}}{T_w - pp}}} \cdot \frac{L}{13.9} \cdot E_{nut} \quad (9)$$

where $M_{(2)}$ is the available food for suspension feeders, F is the unit conversion factor described in Eq. (10), and p_{pp} is the maximum daily primary production

previously reported (5 g C m⁻² d⁻¹; Stockwell, 1989), respectively. The following

term, $\frac{1}{e^{\frac{T-T_{opt}}{T_{w_pp}}}}$, is the temperature limitation for primary production

(dimensionless). The same type of response curve as in benthos (Eq. 4) was used, but a different weighting for temperature limitation (T_{w_pp} in °C) was calibrated

for primary production. $\frac{L}{13.9}$ is the day length (L ; hours) normalized by

maximum day length (13.9 hours) in the area between July and August. This is a scaling factor to represent light limitation resulting from the length of daylight.

E_{nut} is the nutrient limitation (dimensionless) for photosynthesis that includes concentrations of nitrogen (N), silica (Si), and phosphorus (P). The following adjustment is to convert the unit:

$$F = \frac{30}{0.42} \times \frac{10}{100 \cdot d_{(j)}} \quad (10)$$

where $d_{(j)}$ is the water depth (m) and the constants are used to convert from a day to a month (30 days per month), from carbon to dry weight (42% of carbon content per dry weight), and meter to centimeter (100 cm per m). Because suspension feeders are assumed to use the available food source 10 cm above the sediment surface, 10 (cm) is the fraction (i.e., boundary layer) of the entire water column where suspension can reach to feed on primary production carbon.

However, when the water column is well mixed, the thickness of the boundary layer will expand, thus, feeding behaviors of suspension feeder may not be limited to the primary production in the lower 10 cm. To investigate the effects of feeding depth on simulated suspension feeder biomass, a sensitivity test was run. It was found that changes in boundary layer from 10 cm to 200 cm (average depth of the Texas estuaries) made negligible increase (< 2%) in the simulated biomass of suspension feeder in the study area. The final unit for suspension

feeder food availability is measured in $\left(\frac{g \cdot dw}{m^2 \cdot month}\right)$ where dw stands for dry weight.

Nutrient limitation (E_{nut}) for photosynthesis was based on Liebig's law of the minimum, which in this case states that nutrient-dependent photosynthesis occurs at the rate permitted by the most limiting nutrient. In other words, the minimum of the three Michaelis-Menten equations for uptake kinetics of N, Si and P will determine the photosynthetic rate.

$$E_{nut} = MIN\left(\frac{[N]}{[N] + K_N}, \frac{[P]}{[P] + K_P}, \frac{[Si]}{[Si] + K_{Si}}\right) \quad (11)$$

where $[N]$, $[P]$, and $[Si]$ are concentrations (μM) of inorganic nitrogen, phosphorus and silica, respectively. K_N , K_P , and K_{Si} are half-saturation concentration (μM) for nitrogen, phosphorus and silica, respectively. K_N was assumed to be $2.0 \mu M$ (Fisher et al., 1992), and K_P and K_{Si} were assumed to be 0.25 and $1.4 \mu M$, respectively (Tilman and Kilham 1976). Authors used literature values for half-saturation coefficients instead of using conversion based on Redfield ratio (Redfield, 1934) in order to better simulate realistic conditions for nutrient uptake by phytoplankton cells. Therefore, the rate constant determining photosynthesis (the minimum value of Michaelis-Menten kinetics in Eq.11) was determined by not only ambient nutrient concentrations but growth conditions of phytoplankton cells.

A forcing function was derived to simulate day length (number of hours) in order to represent seasonal effect of photosynthesis:

$$L_{(t)} = 12.15849 - 1.755811 \cdot \cos\left(\frac{2\pi \cdot (t)}{12} - 0.2244535\right) \quad (12)$$

where $L_{(t)}$ is day length (h) at time t (month). In general, the photosynthesis-irradiance (P vs. I) relationship reveals hyperbolic response function (e.g., Jassby

and Platt, 1976; Platt and Jassby, 1976; Platt et al., 1980; Neale and Richerson, 1987; Platt et al., 1988). Strictly speaking, it is photo-physiological responses of phytoplankton cells and irradiance that drive changes in primary production, not day length. However, given that day length and average (not day-to-day) irradiance both vary as seasonal harmonics (e.g., cosine function in this study), and should be proportional. Therefore, the simple normalized formulations (Eqs. 9 and 12) should accomplish the general purpose of light limitation.

2.3.3. Predation

Predation is a function of environmental factors (e.g., temperature), benthic biomass, and predator density. In this study, the predation rate ($\text{g dw m}^{-2} \text{ month}^{-1} \text{ individual}^{-1}$) is modeled as a function of temperature and prey benthic biomass, because temperature (Houde, 1987; Houde, 1989; Pepin, 1991; Poulet et al., 1995; Calbet and Agusti, 1999) and prey aggregation (Montagna et al., 1993) are related to the feeding rate of predators.

Predation was simplified as a function of temperature and prey density. Therefore, the term g in Eq. (2) is the temperature- and density-dependent predation rate ($\text{g dw m}^{-2} \text{ month}^{-1} \text{ individual}^{-1}$) and was parameterized as follows:

$$g = 10^{T_{w_pred} \cdot (T - T_{2_{opt}})} \cdot I_v \quad (13)$$

where, T is temperature ($^{\circ}\text{C}$) in both bays, $T_{2_{opt}}$ is an optimal temperature ($^{\circ}\text{C}$) that determines the range of responses, and T_{w_pred} ($^{\circ}\text{C}^{-1}$) is the sensitivity parameter for temperature-dependence. I_v is a modified Ivlev's equation (Ivlev, 1961) by Mayzaud and Poulet (1978) to express the non-linear effect of prey density on predation rate (Eq. 14):

$$I_v = \lambda \cdot B \cdot (1 - e^{-\lambda \cdot B}) \quad (14)$$

where, B is the biomass of the benthic prey in g dw m^{-2} , and λ ($(\text{g dw m}^{-2})^{-1}$) is the parameter for the aggregation effect of prey. Therefore, as T increases, the term g increases to its maximal grazing rate. When benthic biomass (B) is at a very low level the value of term I_v is close to 0, and aggregation effect is nil.

2.4. Model calibration

Model calibration is necessary before actual simulation is conducted because it can set boundaries within which reasonable and localized model behaviors can be produced. The calibration process is based on long-term average values of state variables (i.e., benthic biomass in this study), environmental forcing (day length as a proxy of solar irradiance, temperature, and salinity), and locally-derived (or measured) physiological rate constants (nutrient uptake, grazing, and mortality rate). Long-term time series for state variables and forcing functions are fundamental sources and prerequisite for the model implementation, and in most cases, they are available, but physiological parameters are often rare. Two approaches are frequently used to resolve the unknown parameters: 1) using literature values, and 2) Monte Carlo analysis. In this study, Monte Carlo analyses were used to estimate unknown parameters presented in the previous equations (Eqs. 4 – 14). The Monte Carlo approach used here assigns random numbers to those parameters from defined probability distributions and then a simulation result with the given set of parameters is compared to observations. These steps (i.e., selection of random set – simulation – comparison) are iterated hundreds or thousands of times (e.g., 15,000 times in this study) until the model simulation satisfies a criterion for the model prediction versus data observation comparison (Fig. 5). This process is called optimizing a cost function, which is, in this study, minimization of goodness of fit (GoF). When performing multivariate parameter estimation of nonlinear models a quantitative measure for the difference between simulated results and

observed data can be a cost function (de Hoop et al. 1989). Because the current model has two state variables, the average of these two normalized residuals (difference between model results and observed data, normalized by standard deviation) was used as a Goodness of Fit, and therefore cost function. An 11-year period (April 1988 to October 1999) was used for model calibration. The initial ranges for the parameters were chosen to define upper and lower bounds of the calibration. Fifteen thousand calibration runs were conducted until all parameter ranges were reduced to less than 50% of the initial ranges. Within the initial ranges, the best fit parameter set, which has the minimum normalized residuals, was chosen (Table 3). These best fit values were used as parameters for long-term simulations of benthic macrofauna biomass in the eight Texas estuaries for a 20-year period from April 1988 to October 2008. In this study, each estuary was divided into two bays: the primary bay and the secondary bay (Fig. 1). The model was calibrated for each bay independently.

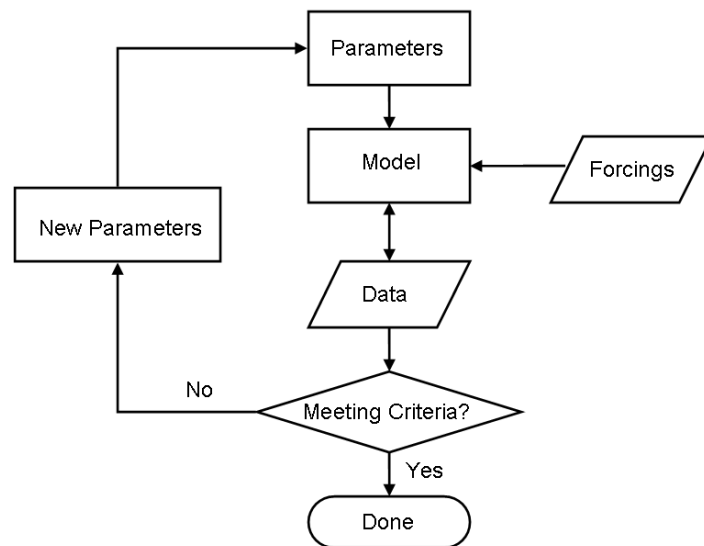


Fig. 5. Schematic of benthic model calibration. Monte Carlo scheme reiterates the loop with newly and randomly chosen new set of parameters for another simulation until model-data comparison meets certain criteria.

SENECA (de Hoop et al., 1989) provides statistical and numerical features to perform the calibration. For interested readers, the developer of SENECA (Netherlands Institute of Ecology – Center for Estuarine and Coastal Ecology) released a new modeling software package *femme* (flexible environment for mathematically modeling the environment) that includes advanced numerical techniques (e.g., adjoint method). For more information, visit <http://www.nioo.knaw.nl/projects/femme/>.

2.5. Model validation

To evaluate the model performance, the percent root mean square (RMS) difference was calculated between model outputs and observations (Eq. 15). An observed data set for the period April 1988 – October 2008 was used for validation of the model.

$$\%RMSD = \sqrt{\frac{\sum \frac{(X_{MOD} - X_{OBS})^2}{N}}{\sum \frac{(X_{OBS})^2}{N}}} \times 100 \quad (15)$$

where X_{MOD} and X_{OBS} are model simulations and observed data respectively. N is the size of the sample (i.e., number of individual data points).

3. Results

3.1. Salinity and temperature differences among estuaries

Significant long-term differences in salinity and temperature can be seen among the estuaries during the calibration period, but not for dissolved oxygen (Table 3). Salinity is significantly different in each estuary (GE = 15.0, LC = 20.1, NC = 28.3, and LM = 37.9). Salinity is significantly lower in the secondary bay than the primary bay in all estuaries, DO is significantly higher, but temperature is the same in primary and secondary bays.

Table 3. Water column characteristics of the bays during the calibration period April 1988 to October 1999. A. Mean (and standard deviation). B. Analysis of variance using a nested model where the water parameter is a function of the estuary and bay nested within estuary and each sampling period is a replicate. Abbreviations: LC=Lavaca-Colorado Estuary, GE=Guadalupe Estuary, NE=Nueces Estuary, LM=Laguna Madre Estuary, LB=Lavaca Bay, MB=Matagorda Bay, USB=upper San Antonio Bay, LSB=lower San Antonio Bay, NB=Nueces Bay, CCB=Corpus Christi Bay, BB=Baffin Bay, ULM=upper Laguna Madre, DF = degrees of freedom, and P = probability.

A.		Salinity(psu)		Temperature(°C)		DO(mg/l)	
Estuary	Bay	Mean	STD	Mean	STD	Mean	STD
LC	Secondary (LB)	15.3	9.0	21.0	6.9	8.0	1.7
LC	Primary (MB)	24.9	4.9	21.4	6.5	7.1	1.9
GE	Secondary (USB)	11.2	8.2	22.1	6.8	8.2	2.0
GE	Primary (LSB)	18.8	9.1	21.8	6.9	8.1	2.0
NE	Secondary (NB)	25.2	8.2	22.9	6.1	7.1	1.5
NE	Primary (CCB)	31.2	4.2	22.6	6.1	6.4	1.8
LM	Secondary (BB)	38.9	9.9	23.0	6.2	6.1	1.6
LM	Primary (ULM)	37.0	8.2	23.7	6.0	8.2	2.1

B.		P		
Source	DF	Salinity	Temp	DO
Estuary	3	0.026	0.0074	0.5044
Bay(Estuary)	4	<0.0001	0.9745	<0.0001
Error DF	378			

3.2. Model validation for the time series (1988 – 2008)

The Monte Carlo technique was used to calibrate the model for the 11-year period (1988 – 1999), and the calibrated parameters in Table 4 represent the best fit values for the Lavaca-Colorado Estuary. Using this calibrated parameter set, the model was validated for periods of analyses given in Table 1. The initial range for temperature was set from 0 to 50 °C, thus some of the estimated temperature parameters are very low or very high. This range reflects a limitation in our method, which is to estimate 19 unknown parameters.

To evaluate the model performance the percent RMS difference was calculated between model outputs and observations (Eq. 15). The percent RMS differences between the simulated and observed biomass for deposit and suspension feeder in all eight bays ranged from 62.7% (deposit feeder in Lower San Antonio Bay) to 116.2% (deposit feeder in Nueces Bay). Table 5 lists percent RMS differences between the simulated and observed biomass for two benthic groups along the Texas estuaries. Overall, the low percent RMS values indicate the model performance with calibrated parameters was found to be successful.

Table 4. Best fit parameter values from the Monte Carlo analysis for the eight Texas bays. The calibration is based on for the eleven-year database: 1988 – 1999, and the parameters are defined in Equations (2 – 14). Abbreviations: Param=Parameter; Ben=Benthos; LB=Lavaca Bay, MB=Matagorda Bay, US=Upper San Antonio Bay, LS=Lower San Antonio Bay, NB=Nueces Bay, CC=Corpus Christi Bay, BB=Baffin Bay, LM=Upper Laguna Madre.

Parameter	Benthic Group	Best Fit Values								Units
		LB	MB	US	LS	NB	CC	BB	LM	
$r_{(1)}$	Dep	5.116	5.311	5.119	5.084	5.200	5.337	5.104	7.009	month ⁻¹
$c_{(1)}$	Dep	30.799	32.705	61.449	42.376	30.329	30.567	38.320	31.154	mg dw m ⁻²
$r_{(2)}$	Epi	7.844	5.172	5.862	6.257	5.314	12.816	5.426	5.445	month ⁻¹
$c_{(2)}$	Epi	97.070	86.395	77.920	99.701	50.433	67.779	95.759	96.316	mg dw m ⁻²
$T1_{opt}$	Dep	49.422	46.503	49.261	0.561	47.950	49.747	26.600	3.274	°C
$T1_{opt}$	Epi	0.792	2.549	0.592	2.137	37.460	42.517	25.957	0.294	°C
T_{w_ben}	Dep, Epi	30.244	33.195	32.272	30.230	35.267	35.046	31.193	47.873	°C
S_{opt}	Dep	20.140	36.410	38.222	39.349	38.839	39.697	33.947	21.902	psu
S_{opt}	Epi	0.181	5.433	7.301	19.275	7.159	11.162	0.249	5.319	psu
S_{w_ben}	Dep, Epi	30.303	31.482	59.369	53.532	58.455	57.186	57.945	30.310	psu
K_M	Dep	80.593	89.934	88.873	31.682	88.284	0.891	96.642	31.628	g dw m ⁻²
K_M	Epi	99.284	67.730	0.515	30.087	0.646	17.131	61.948	78.109	g dw m ⁻²
$c_{p_{pom}}$	Dep	1444.6	1775.2	3962.2	2292.6	2135.5	1847.2	2821.1	1678.3	g dw m ⁻²
p_{sed}	Dep	^b 16.4	16.4	16.4	16.4	16.4	16.4	16.4	16.4	g dw (35.4 cm ²) ⁻¹
P_{pp}	Epi	5.0	5.0	5.0	5.0	5.0	5.0	5.0	5.0	g C m ⁻² d ⁻¹
T_{w_pp}	Epi	48.967	49.737	17.501	16.338	16.145	40.661	17.460	19.780	°C
dK_N	Epi	2.0	2.0	2.0	2.0	2.0	2.0	2.0	2.0	μM
eK_P	Epi	0.250	0.250	0.250	0.250	0.250	0.250	0.250	0.250	μM
$^fK_{Si}$	Epi	1.4	1.4	1.4	1.4	1.4	1.4	1.4	1.4	μM
T_{w_pred}	Dep	0.006	0.115	0.341	0.123	0.007	0.100	0.051	0.684	°C
$T2_{opt}$	Dep	22.232	38.921	31.211	30.789	24.130	49.691	46.939	32.990	°C
T_{w_pred}	Epi	0.232	0.080	0.225	0.057	0.009	0.062	0.474	0.134	°C
$T2_{opt}$	Epi	33.368	32.790	28.060	47.384	9.896	37.575	28.476	36.308	°C
λ	Dep	0.035	0.218	0.155	0.059	0.034	0.719	0.371	0.829	(g dw m ⁻²) ⁻¹
λ	Epi	0.973	0.156	0.043	0.366	0.069	0.649	0.141	0.134	(g dw m ⁻²) ⁻¹

^aDep and Epi represent deposit and suspension feeders, respectively

^bAverage dry weight of the whole sediment (g dw) per core sample with area of 35.4 cm² (Montagna, pers. comm.)

^c P_{pom} values were based on P_{poc} measurements from CHN analyzer (see the text)

^d K_N (Fisher et al., 1992); ^e K_P (Tilman and Kilham 1976); ^f K_{Si} (Tilman and Kilham 1976)

Table 5. The percent root mean square (RMS) difference between observed and simulated benthic biomass in the bays during the long-term simulation period of analyses (see Table 1). Abbreviations: LB=LavacaBay, MB=Matagorda Bay, US=Upper San Antonio Bay, LS=Lower San Antonio Bay, NB=Nueces Bay, CC=Corpus Christi Bay, BB=Baffin Bay, LM=Upper Laguna Madre.

Bay	Benthic group	April 1988 – October 2008 (for long-term simulation)
		% Root Mean Square
LB	Deposit feeders	101.2
	Suspension feeders	93.6
MB	Deposit feeders	67.9
	Suspension feeders	85.7
US	Deposit feeders	69.5
	Suspension feeders	83.1
LS	Deposit feeders	73.6
	Suspension feeders	97.7
NB	Deposit feeders	61.0
	Suspension feeders	95.4
CC	Deposit feeders	116.1
	Suspension feeders	88.0
BB	Deposit feeders	63.0
	Suspension feeders	77.0
LM	Deposit feeders	103.2
	Suspension feeders	74.2

3.3. Model simulation results

The simulations of benthic biomass are based on the best fit parameters from the calibration of the period (1988 to 1999). All simulations were run during the period of analyses (see Table 1) and results were compared to observed benthic macrofauna biomass data (Figs. 6, 8, 10 and 12).

3.3.1. Lavaca-Colorado Estuary (Lavaca and Matagorda Bay)

The pattern for the simulations of deposit feeders in Lavaca Bay was increases from 1987 to 1993 followed by a dramatic drop in biomass with the lowest biomass concentration occurring between 1994 and 1996 (Fig. 6a). After the year 2000, the biomass showed signs of slow increase. Simulations of suspension feeder biomass in the same bay followed the same pattern as the deposit feeder biomass, and had a trend of a slower increase in biomass after the year 2000 (Fig. 6b). In Matagorda Bay, simulations of deposit feeders showed a trend having low biomass during the period 1994 to 2000 and increased biomass in 2001 and 2002 (Fig. 6c). Simulations of suspension feeders for Matagorda Bay showed a similar trend to that of deposit feeders (Fig. 6d); having low biomass during high the period 1994 to 2000 and increased biomass around 2001 and 2002, but the model failed to capture significantly increased biomass in 2008, which was as high as that in 1989, being an order of magnitude higher than low biomass during the period 1994 to 2000.

The simulations for both bays and each feeding group fit the observed data relatively well during the entire period, 1988 to 2008 (Fig. 6, Table 5). It is also noteworthy that the trends of the prediction over time fit the trends in the observed biomass for both bays. However, the model still needs improved performance (i.e., minor tuning processes) because a few segments of the time period showed large deviations. For example, the worst fit happened to the deposit feeders in Lavaca Bay (101.2%), which shows large discrepancies during

the period of 1994 to 2000 when the model predicts rather low and constant biomass, and 2004 to 2008 when the model overestimates the biomass compared to the actual data (Fig. 6a). Similarly, simulations of suspension feeders on the same bay were underestimated during entire period (Fig. 6b). The deviation in Lavaca Bay seems to be attributed to the misfit for the period from 1994 to 2000 when predators' biomass was high (Fig. 6a). This is because an increase in predator populations during this period, particularly blue crabs (see Fig. 7a), did not have much predation impact on prey, which caused prey populations to remain relatively high during this period; whereas, the model simulations were still strongly influenced by the increased number of predators during this period when numbers of blue crabs caught in Lavaca Bay have increased (Fig. 7a). Same is true for the period from 2004 to 2008 when predators' biomass was low (Figs. 6a and 7a). The model simulations were still strongly influenced by the reduced predation pressure during this period when numbers of blue crabs caught in Lavaca Bay remained low (Fig. 7a).

In Matagorda Bay, the simulated results (Figs. 6c, d) had a coupled trend of both deposit and suspension feeders with a decreasing trend during the high predation period (1994 – 2000), and an increasing trend during times of low blue crab biomass (Fig. 6b). Blue crab abundance in Matagorda Bay also had a similar trend of that in Lavaca Bay, having higher abundance during the period 1994 and 2000 and lower abundance in the period after 2000 (Fig. 6b). The simulations for the deposit feeders fit the data best (67.9%, Table 5), having low biomass during the high predation period (1994 – 2000) and high biomass when the number of predators starts decreasing in the 2001 and 2002 period (Fig. 6b). The higher variance for suspension feeders in Matagorda Bay can be attributed to underestimated spiky biomass in 1989 and 2008 (Fig. 6d).

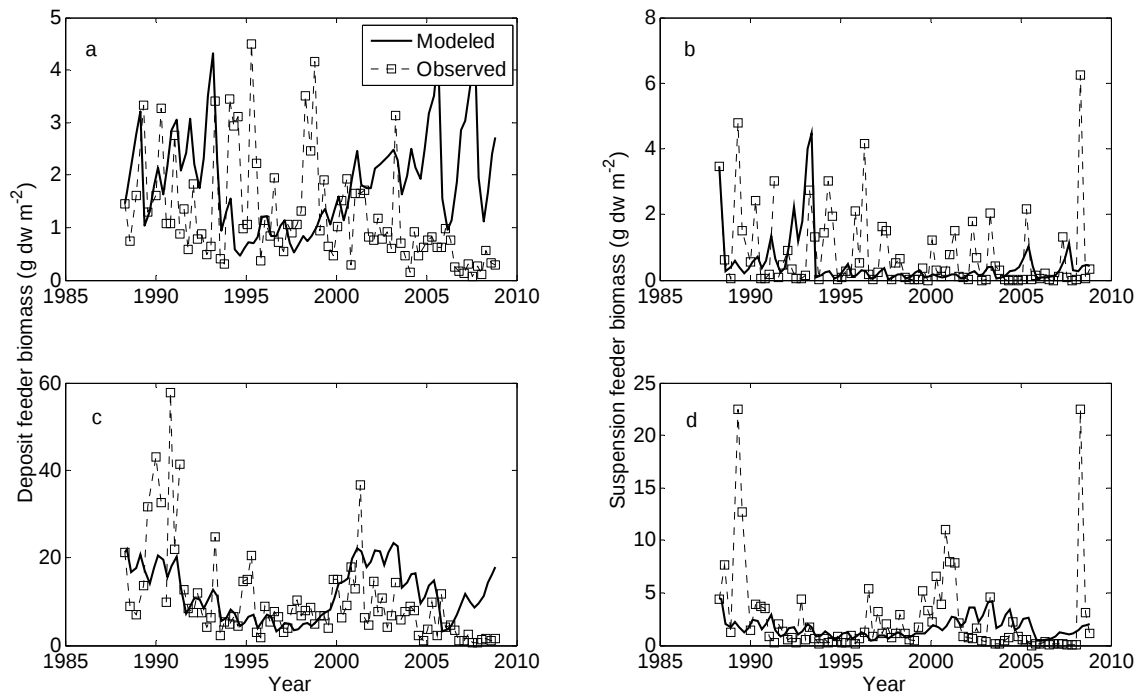


Fig. 6. Comparison between observed (dashed line with open squares) and modeled (solid lines) results in Lavaca and Matagorda Bay for the period 1988-2008. Each panel represents comparison results for (a) deposit and (b) suspension feeder biomass in Lavaca Bay, and (c) deposit and (d) suspension feeder biomass in Matagorda Bay.

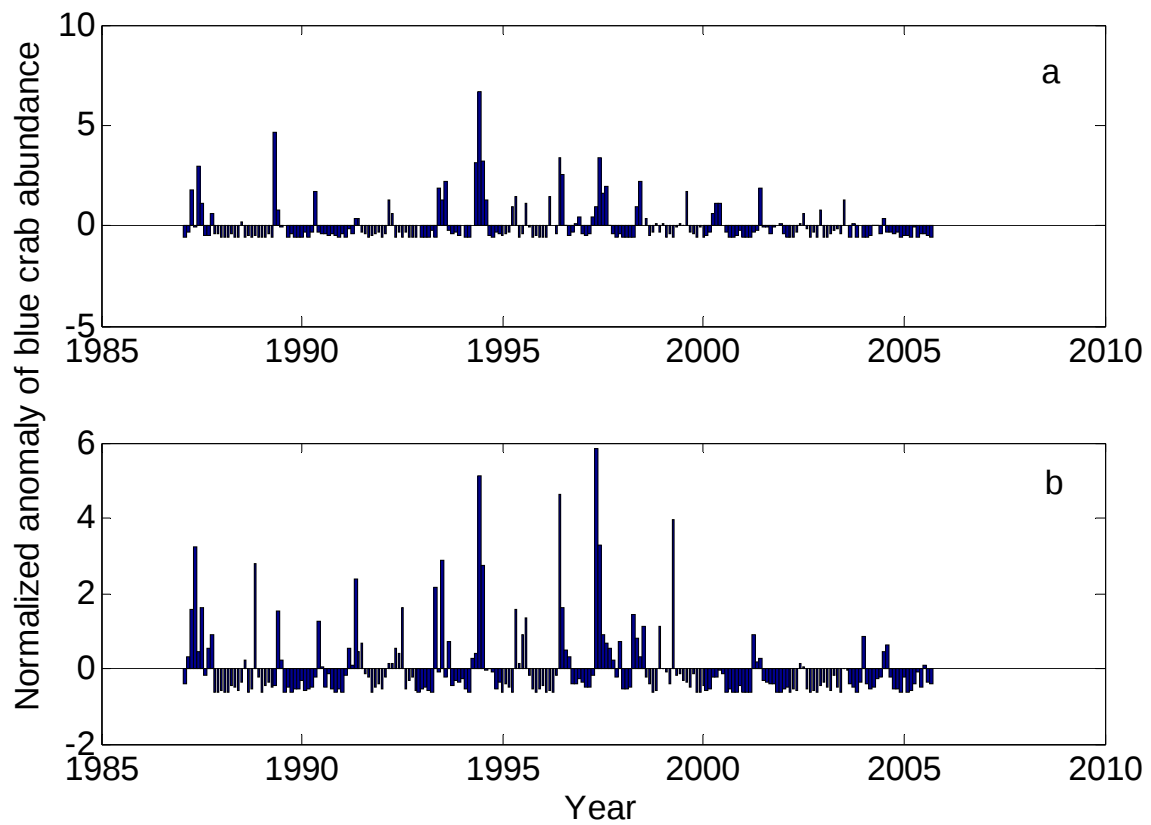


Fig. 7. Normalized anomaly of blue crab abundance. Positive and negative values represent above and below long-term average, respectively. Anomaly was normalized by standard deviation. Blue crab population data (1987 – 2006) were collected by TPWD for (a) Lavaca Bay and (b) Matagorda Bay.

3.3.2. *Guadalupe Estuary (upper and lower San Antonio Bay)*

Simulations of deposit feeders in upper San Antonio Bay revealed low biomass at beginning, followed by a slow increase between 1995 and 1998 (Fig. 8a). Simulated biomass decreased during the period of 1998 to 1999, and increased in 2000. Simulations of suspension feeder biomass in the same bay followed the similar pattern as the deposit feeder biomass, but failed to capture peaky biomass during 1994 to 1996 (Fig. 8b). In lower San Antonio Bay, simulations of deposit feeders were successful during the period 1993 – 2000 (Fig. 8c). However, like simulations of deposit feeders in upper San Antonio Bay, the model could not simulate high biomass during 1988 to 1991. Simulations of suspension feeders for the same bay showed a similar trend to that of suspension feeders in upper San Antonio Bay, not being able to capture peaky biomass during 1993 to 1994 (Fig. 8d).

The statistics reveal that the simulations for both bays and each feeding group fit the observed data relatively well showing 76 to 98% of RMS difference (Table 5), but again the model still needs improved performance for early period of simulations (e.g., deposit feeders) and for a few spiky period that showed large deviations (e.g., suspension feeders). Like Lavaca-Colorado Estuary, the deviation of deposit feeders in both upper and lower San Antonio Bay is attributed to the misfit for the early period (e.g., the period of 1988 – 1994) when predators' biomass was high (Figs. 9a and b). Again, the model simulations were still strongly influenced by the increased number of predators during this period. For suspension feeders in the two bays the model was not able to reproduce significantly increased biomass during mid-1990's.

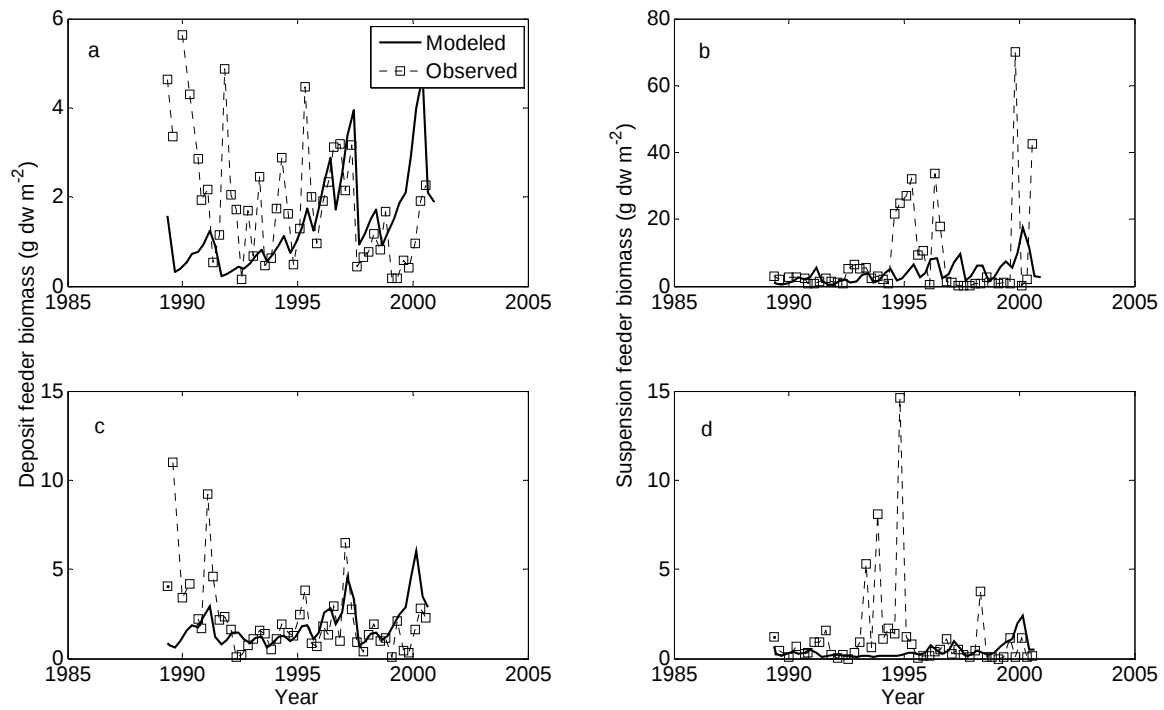


Fig. 8. Comparison between observed (dashed line with open squares) and modeled (solid lines) results in San Antonio Bay for the period 1989-2008. Each panel represents comparison results for (a) deposit and (b) suspension feeder biomass in upper San Antonio Bay, and (c) deposit and (d) suspension feeder biomass in lower San Antonio Bay.

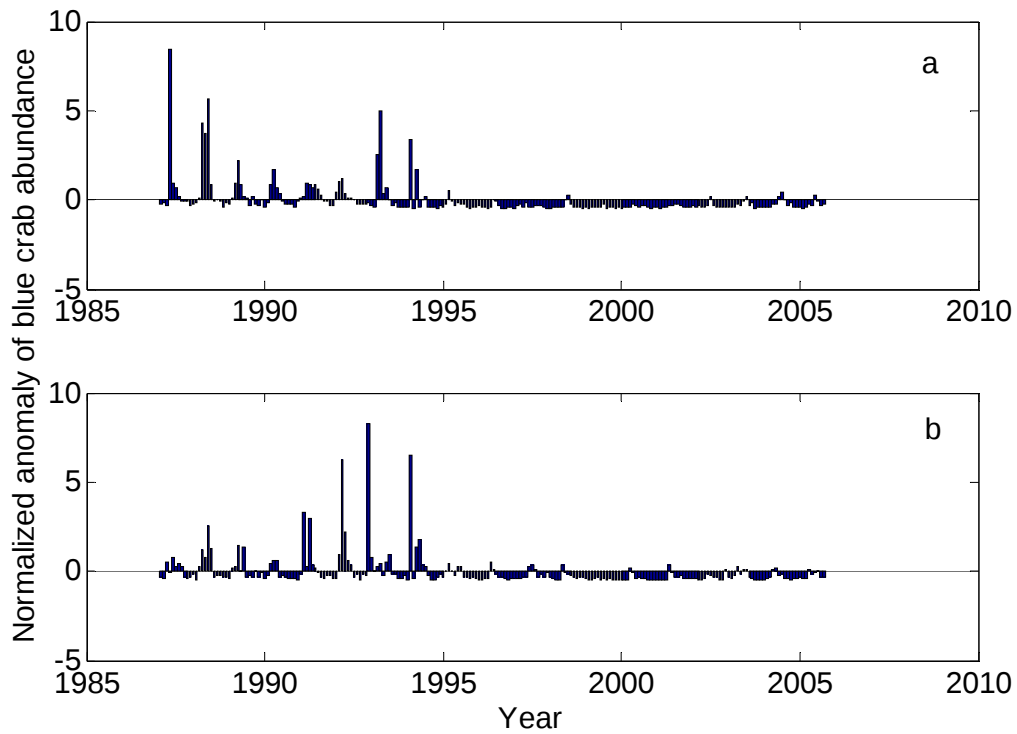


Fig. 9. Normalized anomaly of blue crab abundance. Positive and negative values represent above and below long-term average, respectively. Anomaly was normalized by standard deviation. Blue crab population data (1987 – 2006) were collected by TPWD for (a) upper and (b) lower San Antonio Bay.

3.3.3. Nueces Estuary (*Nueces and Corpus Christi Bay*)

Simulations of deposit feeders in Nueces Bay were successful overall, capturing a trend of variation shown in data from early to late 1990's. However, the model simulated biomass was too low post-2000 compared to observations (Fig. 10a). Both simulations and observations of suspension feeder biomass in the same bay revealed similar patterns to those in San Antonio Bay (especially, lower San Antonio Bay), not being able to capture significantly high biomass during 1993 to 1994 (Fig. 10b). In Corpus Christi Bay, simulations of deposit feeders were successful during the period 1989 to 1995 (Fig. 10c). However, like simulations of deposit feeders in Nueces Bay, the model started showing quite high deviation for the rest of the simulation period (1995 – 2002). Simulated suspension feeders seems to fit the data fairly well in the same bay (Fig. 10d), but the model could not simulate biomass increase in 1991.

The percent RMS difference between the model results and data was 62.7 and 116.2% for deposit feeders in Nueces and Corpus Christi Bay, respectively; and suspension feeders of Nueces and Corpus Christi Bay revealed 95.8 and 89.2 % RMS difference (Table 5). The largest discrepancy in the deposit feeders of Corpus Christi Bay may be explained by the period when predators' biomass was remaining low (1995 – 2002), which drove model to estimate high biomass, as a result (Fig. 11b). However, this was not the case for suspension feeders in the same bay, and considering the fact that these two functional groups have the same predators, there should be more plausible explanation than predation. This will be discussed later.

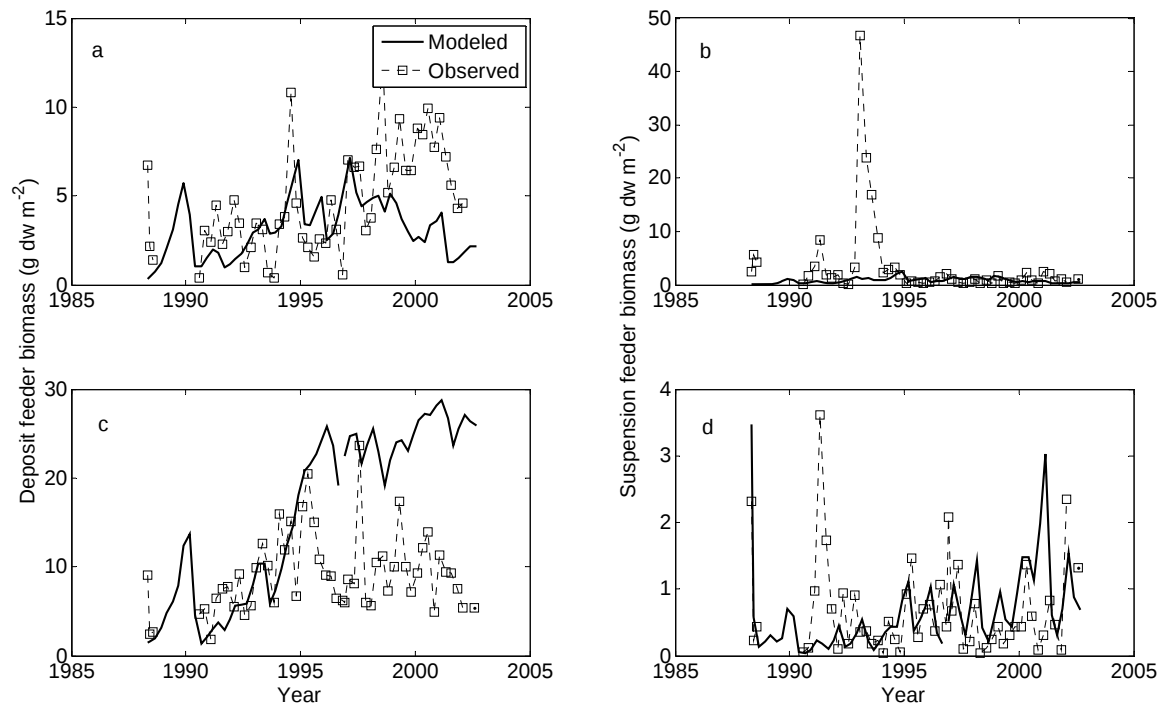


Fig. 10. Comparison between observed (dashed line with open squares) and modeled (solid lines) results in Nueces and Corpus Christi Bay for the period 1989-2008. Each panel represents comparison results for (a) deposit and (b) suspension feeder biomass in Nueces Bay, and (c) deposit and (d) suspension feeder biomass in Corpus Christi Bay.

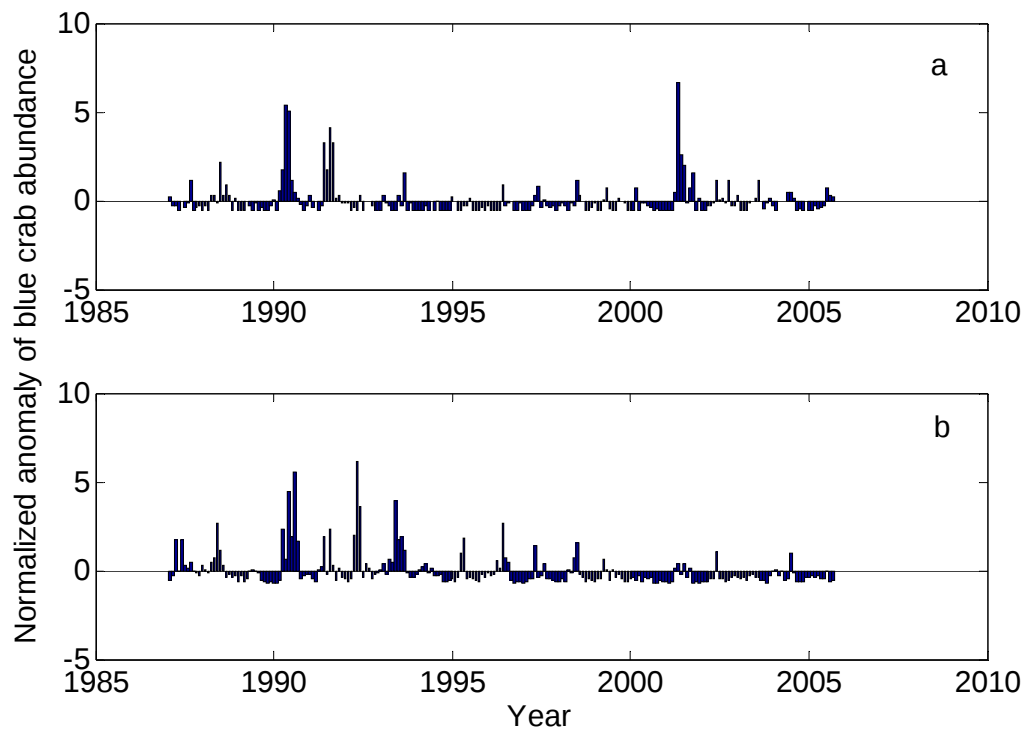


Fig. 11. Normalized anomaly of blue crab abundance. Positive and negative values represent above and below long-term average, respectively. Anomaly was normalized by standard deviation. Blue crab population data (1987 – 2006) were collected by TPWD for (a) Nueces and (b) Corpus Christi Bay.

3.3.4. *Laguna Madre Estuary (Baffin Bay and upper Laguna Madre)*

The percent RMS difference between simulation and observation in Baffin Bay was 63% and 77% for deposit and suspension feeders, and in upper Laguna Madre values were 98% and 68% for deposit and suspension feeders, respectively. Results revealed that model performance was overall satisfactory. Specifically, model simulations for deposit feeders in Baffin Bay and suspension feeders in upper Laguna Madre were able to reproduce the fluctuating trend for entire simulation period, whereas, simulated results for deposit feeders from upper Laguna Madre failed to reproduce the extremely high values in during 1992 to 1993 (Fig. 12). Like suspension feeders in Corpus Christi Bay, time series recorded extreme values during 1992 to 1993 and they were on the order of magnitudes higher than data collected from other period. Interestingly, this event coincides with elevated predator abundances (Fig. 13.b), indicating that predation may not be the only factor driving benthic biomass changes.

The simulations of benthos biomass for deposit-feeders and suspension feeders in all bays were successful overall and fit of the simulation results revealed by the percent RMS difference ranged from 63% (deposit feeders in Nueces Bay) to 116% (deposit feeders in Corpus Christi Bay) (Table 5). The simulations fit well for most bays over the period of simulations, except for events of extremely high values, which cannot be explained given model structure. We leave this to future investigation.

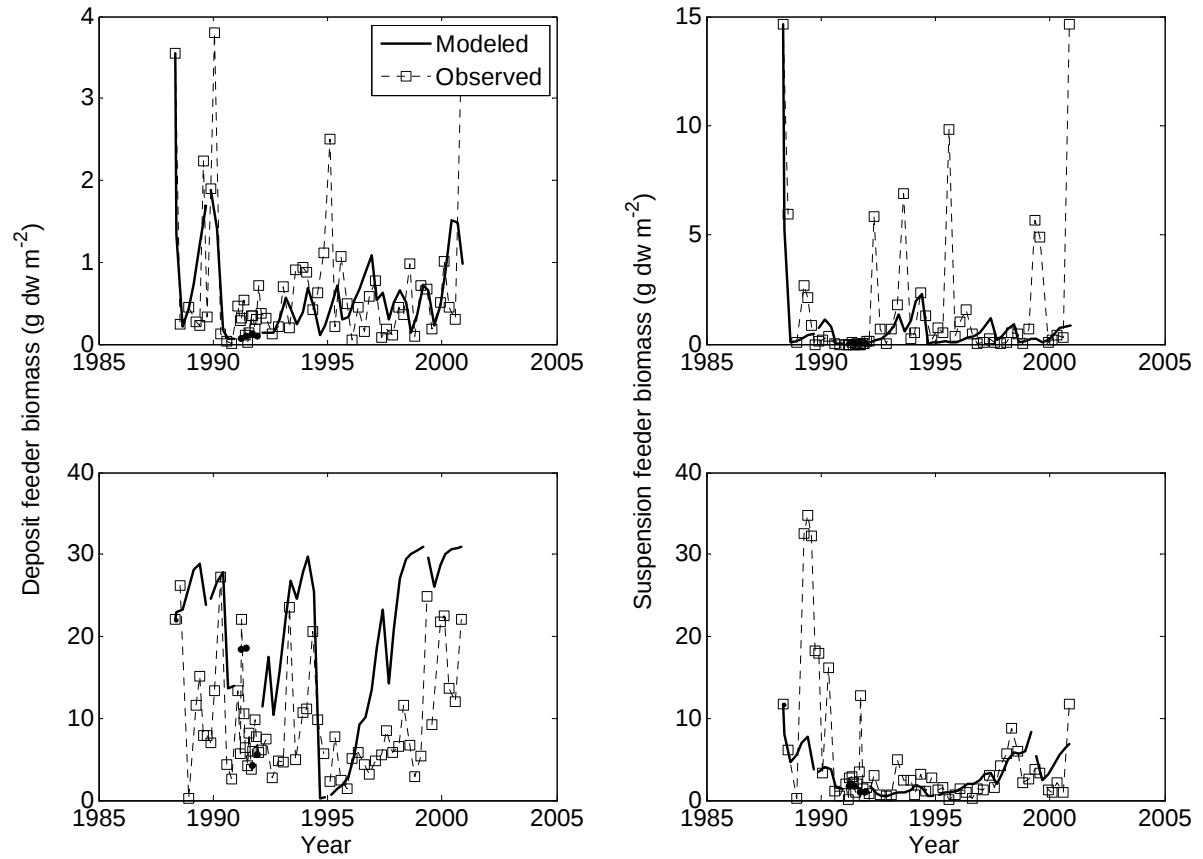


Fig. 12. Comparison between observed (dashed line with open squares) and modeled (solid lines) results in Baffin Bay and upper Laguna Madre for the period 1989-2008. Each panel represents comparison results for (a) deposit and (b) suspension feeder biomass in Baffin Bay, and (c) deposit and (d) suspension feeder biomass in upper Laguna Madre.

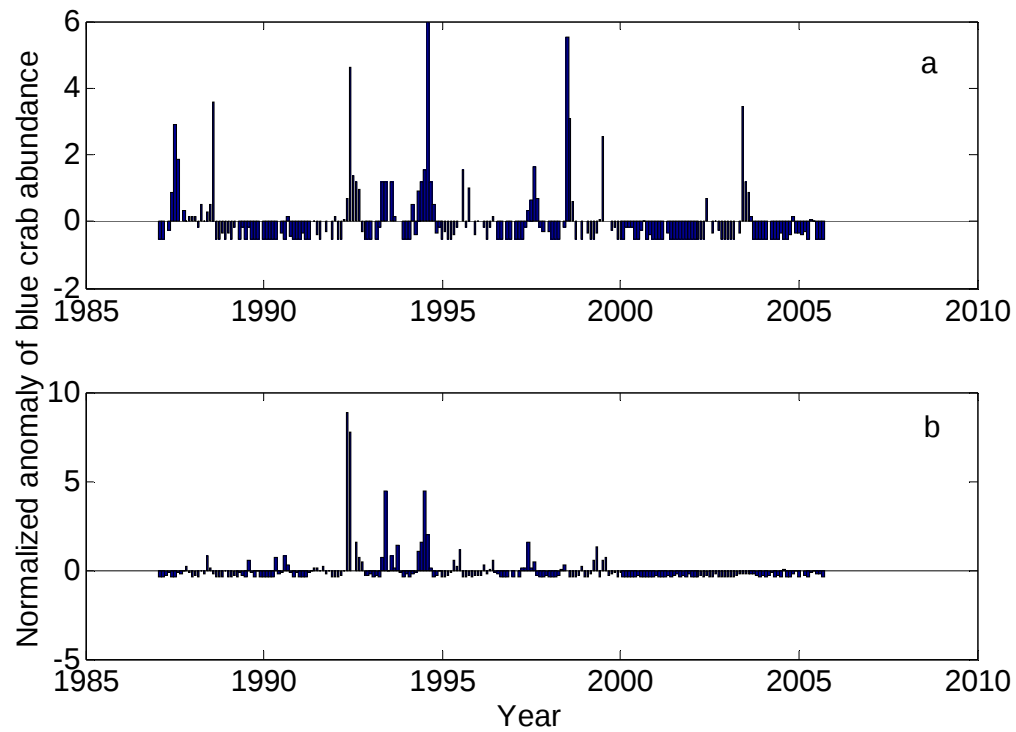


Fig. 13. Normalized anomaly of blue crab abundance. Positive and negative values represent above and below long-term average, respectively. Anomaly was normalized by standard deviation. Blue crab population data (1987 – 2006) were collected by TPWD for (a) Baffin Bay and (b) upper Laguna Madre.

4. Discussion

4.1. Ecosystem modeling and goodness of fit

Ecosystem models are representations of underlying mechanistic relationships among ecological components and processes. Ideally, they reduce ambiguity and describe complexity with maximum parsimony. Models in ecology are useful because of the inherent complexity of ecological relationships, the characteristic variability in ecological systems, and the apparently unpredictable effects of deliberate modification of systems by man. It is difficult to understand benthic dynamics from empirical or static modeling analyses alone (e.g. multivariate statistical methods). The concept that benthos are an isolated subsystem, governed by internal interactions and “key species” is not sufficient to explain the heterogeneity of benthos community dynamics in closely related sites. However, a model can incorporate spatial variability to provide insights into the dynamics and interactions of benthic populations within an ecosystem, or to predict long-term effects of those interactions.

Modeling of an ecosystem can start from a qualitative conceptual model. The conceptual model is largely theoretical and heuristic. The purposes for modeling ecosystems can range from developing simple conceptual models to provide a general understanding of system behavior, to detailed realistic applications aimed at evaluating specific policy proposals. It is not possible to judge this whole range of models by the same criteria. At least three criteria are necessary: realism, precision, and generality. Unfortunately, no single model can maximize all three. The conceptual model has high generality, but low realism and low precision. A quantitative model, however, can provide the realism and precision, and test hypotheses drawn from the conceptual model. Usually, a quantitative model requires a long-term data set to calibrate the model. In addition, independent data sets are also needed for model corroboration or validation before these models are used for extensive predictions.

In general, development and implementation of an ecosystem model are not trivial processes because they involve high level of uncertainty. de Hoop et al. (1989) did very nice and brief review on the concept of uncertainty. Major sources of uncertainty include (O'Neill and Gardner, 1979; Walters, 1986): 1) uncertainty of model structure; 2) uncertainty in the parameters, and 3) uncertainty in the inputs. Uncertainty of model structure is almost impossible to figure out because the fundamental governing equations are not known. Uncertainty of the parameters is also hard to quantify when model complexity is high because many parameters are known as a range. In addition, not so many parameters are available from direct measurements from laboratory or other literature. The uncertainty from the input data (e.g. state variables, forcing functions, and other parameters) involves errors in observations or unexpected and abrupt changes in forcing functions.

In this study, we presented results of secondary production in Texas lagoonal estuaries, simulated by a bioenergetic model. Given uncertainties of governing equations, large number of fitted parameters (therefore large uncertainty), and limited number of data, authors have to admit there are limitations of this approach for extensive applications. For example, the percent root mean square difference (%RMSD), which is an indicator of model performance, revealed the range from 61 to 116% between observed and simulated biomass. Compared to primary production models that can estimate primary production within 50 to 60% of %RMSD (e.g., Jung et al., 2007; Friedrichs et al., 2009) these results are somewhat higher indicating poorer performance of the presented model than those primary production models. In general modeling benthic secondary production is challenging because the benthic food web is complex, and because secondary production rate is not a simple function of physico-chemical variables, unlike primary production. Thus, considering the complexity of processes, authors find that the performance of the presented model was found to be promising. The results are also comparable to

other process-based ecosystem models. Two examples of process-based ecosystem models are given here as references for skill assessment: Morales et al. (2005); and Xu and Hood (2006). These models also show results as high %RMSD (10-200%) as ours (see Figs. 1 and 2).

4.2. Simulated benthic biomass

Overall, the heuristic approach presented in this study was found to be successful in simulating benthic biomass. However, the model still needs to be improved by: 1) fixing potential errors resulting from parameterizations, i.e., there are too many unknown parameters estimated from a limited number of observations; 2) including additional environmental variables that could be driving carbon production or benthic growth; 3) modifying governing equations (e.g., including additional state variables for nutrients, chlorophyll *a*, predators, etc.); and most importantly, 4) changing the model structure (e.g., trophic relationships and benthic-pelagic coupling).

Understanding and modeling predation is likely the most important improvement needed. For example, actual observations of benthos biomass in Lavaca Bay (Figs. 6a, b) show little indication of decrease despite the increased predation pressure during the period of 1994 and 2000 (Fig. 7a). As mentioned earlier, black drum, red drum, and blue crab were selected as the main predators for the model. Blue crab was the dominant species found in the TPWD data set during the study period, and the density of the fish species was negligible. The density and impact of fish may not be adequately represented. Perhaps blue crabs are not a top-down controller of benthic biomass in the Lavaca Bay ecosystem, and there are other predators for which we have no data. Additional observations support this hypothesis. Benthic biomass in Lavaca Bay revealed little or no trend of increase despite the decreased predator abundance post-2000 (Fig. 7a). This indicates that another environmental factor or other predators not

included in the model might have caused benthic prey density to remain. In contrast, the Matagorda Bay data (Figs. 6c and d) indicate coupled effects between benthic prey and predators. Low abundance in predators matches high biomass in benthic prey during two periods: 1988 to 1994 and 2000 to 2005. The period of high predation during 1994 to 2000 (Fig. 7b) has decreased benthic biomass. Although the model was able to capture fluctuating trend in Matagorda Bay, the magnitude of biomass change due to predation still needs improvement. In addition, experiments are needed to better explain the trophic relationship between prey and predators in this ecosystem, and this will guide additional data needs and modification of the model structure in the future to achieve more realistic model behavior.

4.3. Salinity/nutrients change scenario and benthic responses

Benthic ecosystem responses to different climate scenarios were investigated by using sensitivity analyses for the salinity and nutrient input parameters, which should change with changing rain, runoff, and inflow. Salinity does vary quite a bit over the long-term (Fig. 14). However, it is also obvious that the direction of salinity change over time is similar in the different estuaries (Fig. 14) because of the wet and dry cycle driven by El Niño (Tolan 2007).

The long term means of the salinity and chlorophyll a was different in different estuaries, but the long-term mean of nutrients was not different (Table 6). The secondary bays always had higher nutrient levels than primary bays.

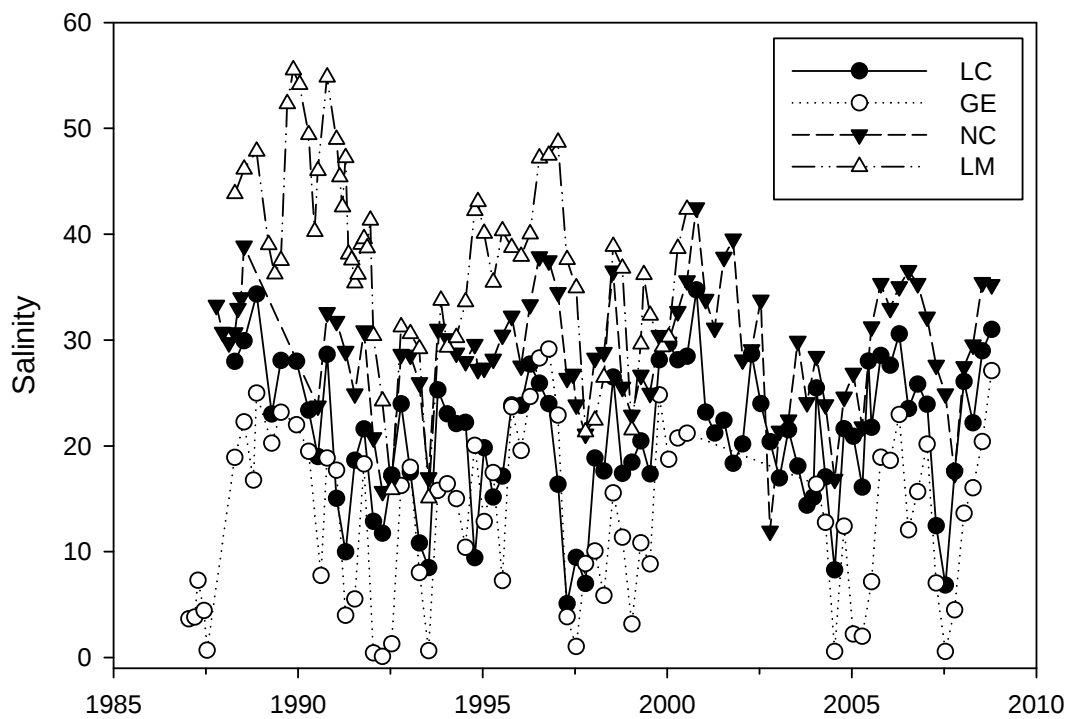


Fig. 14. Estuary-wide salinity variability of time. Abbreviations: LC=Lavaca-Colorado Estuary, GE=Guadalupe Estuary, NE=Nueces Estuary, LM=Laguna Madre Estuary.

Table 6. The long-term water column characteristics in Texas bays. Means were average over stations-bays-dates, and then bays-dates, the number (n) is for bay-date combinations. A) Hydrography. B) Nutrients. C) Analysis of variance using a nested model where the water parameter is a function of the estuary and bay nested within estuary and each sampling period is a replicate. Abbreviations: LC=Lavaca-Colorado Estuary, GE=Guadalupe Estuary, NE=Nueces Estuary, LM=Laguna Madre Estuary, LB=Lavaca Bay, MB=Matagorda Bay, USB=upper San Antonio Bay, LSB=lower San Antonio Bay, NB=Nueces Bay, CCB=Corpus Christi Bay, BB=Baffin Bay, ULM=upper Laguna Madre, n = number, DF = degrees of freedom, and P = probability.

A.			Salinity (psu)		Temperature (°C)		Chl a (mg/l)	
Estuary	Bay	n	Mean	STD	Mean	STD	Mean	STD
LC	Secondary (LB)	82	15.8	8.8	21.8	6.8	8.1	5.2
LC	Primary (MB)	82	24.5	5.8	22.3	6.6	7.8	4.5
GE	Secondary (USB)	69	10.2	7.3	22.8	6.3	12.4	11.6
GE	Primary (LSB)	69	18.0	9.1	22.6	6.4	10.3	9.0
NE	Secondary (NB)	79	25.3	9.0	22.9	6.3	7.0	5.5
NE	Primary (CCB)	80	31.4	4.7	22.7	6.2	4.8	2.9
LM	Secondary (BB)	62	38.3	10.0	23.1	6.0	20.0	16.6
LM	Primary (ULM)	61	37.0	8.1	23.7	5.9	14.2	10.2

B.			NO _x (uM)		NH ₄ (uM)		PO ₄ (uM)		SiO ₄ (uM)	
Estuary	Bay	n	Mean	STD	Mean	STD	Mean	STD	Mean	STD
LC	Secondary (LB)	70	5.4	9.5	2.9	5.7	2.2	6.2	107.8	67.6
LC	Primary (MB)	71	2.9	4.1	1.9	1.9	1.1	0.8	61.9	39.2
GE	Secondary (USB)	60	22.0	21.2	3.1	4.1	3.2	2.7	136.1	71.6
GE	Primary (LSB)	60	6.0	12.4	1.8	2.2	1.7	1.5	105.5	66.6
NE	Secondary (NB)	72	3.3	4.0	2.3	2.9	1.7	1.0	109.1	74.2
NE	Primary (CCB)	72	1.1	1.9	1.5	1.8	0.7	0.6	48.5	40.9
LM	Secondary (BB)	43	1.8	1.9	5.3	5.4	1.1	0.6	92.5	40.5
LM	Primary (ULM)	43	1.3	3.5	2.1	1.8	0.9	0.7	77.7	47.4

C.		P				
Source	DF	Chl	NO _x	NH ₄	PO ₄	SiO ₄
Estuary	3	0.0201	0.2506	0.5873	0.3782	0.5744
Bay(Estuary)	4	0.2070	<0.0001	<0.0001	0.0008	<0.0001
Error DF		233	301	463	467	466

The simulated percent change in deposit (filled circles) and suspension feeder (open circles) biomasses reveal different responses caused by changes in salinity and nutrients in estuarine systems along the Texas Gulf Coast (Fig. 15). The change in benthic biomass was simulated as driven by a percent change of mean salinity (-50% to 50%) and corresponding nutrients (based on the salinity-nutrients relationships in Fig. 2) In Figure 15, the panels a, b, c and d represent benthic responses from Lavaca-Colorado, Guadalupe, Nueces and Laguna Madre Estuary, respectively. The error bars represent the standard error. The long-term mean salinity for all bays is listed in Table 6. The long-term mean biomass for deposit and suspension feeders were 1.24 and 0.77 g dw m⁻² for Lavaca Bay; 10.01 and 2.32 g dw m⁻² for Matagorda; 88 and 7.89 g dw m⁻² for upper San Antonio; 2.18 g dw m⁻² and 1.19 g dw m⁻² for lower San Antonio Bay; 5.09 and 3.33 g dw m⁻² for Nueces; 9.76 and 0.67 g dw m⁻² for Corpus Christi Bay; 0.55 and 1.36 g dw m⁻² for Baffin Bay; and 40.02 g dw m⁻² and 2.61 g dw m⁻² for upper Laguna Madre, respectively.

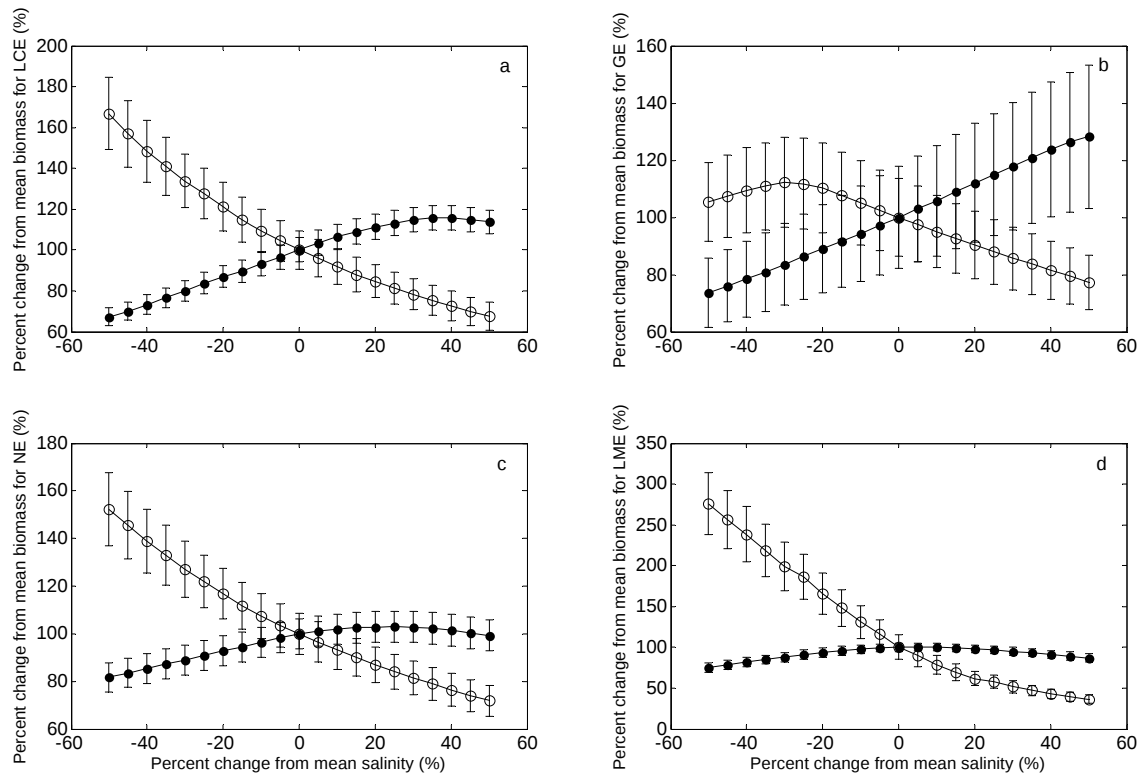


Fig. 15. Percent change from mean benthic biomass simulated with percent change of mean salinity (–50% to 50%) and corresponding nutrients for deposit (filled circles), and suspension feeder biomass (open circles) in Lavaca-Colorado (a); Guadalupe (b); Nueces (c); and Laguna Madre Estuary (d) along the Texas Gulf Coast. The error bars represent the standard error.

Kalke and Montagna (1991) investigated the effects of freshwater inflow on macrobenthos in the Lavaca Bay. A high freshwater inflow rate caused low salinity species to populate the area, and, therefore, it was determined that freshwater is necessary in the upper portion of the bay to induce recruitment of low salinity species. Following an inflow event, Chironomid larvae (suspension feeders) and *Hobsonia florida* (polychaete) increased in density, as both prefer lower salinity environments. In contrast, the mollusks, *Streblospio benedicti* and *Mediomastus californiensis* (deposit feeders), increased in benthic biomass during periods of low freshwater inflow (Kalke and Montagna 1991). In upper Rincon Bayou, near where it connects to the Nueces River, it was found that biomass, abundance and diversity of macrofauna are low at salinities below 10, increase to

salinities of about 20-25, and then decrease (Montagna et al. 2002b). Biomass blooms were also found to occur in the lower part of Rincon Bayou where it connects to Nueces Bay after inflow events (Palmer et al. 2002). These findings from the previous studies are consistent with the results of the sensitivity tests conducted in the present study, which, overall, shows that salinity increase combined with nutrient decreases caused deposit feeder biomass to increase and suspension feeder to decrease (Fig. 15).

It was also found that chlorophyll *a* concentrations increased in Lavaca Bay during high inflow rates, and this indicates that primary production is enhanced by increased nutrients loadings in discharged freshwater (Kalke and Montagna 1991). As salinity increased during periods of low inflow, the chlorophyll *a* concentration decreased due to reduced nutrient loadings and elevated grazing pressure by the growing population of mollusks (Kalke and Montagna 1991). A similar finding occurred in Baffin Bay during the brown tide bloom (Buskey et al. 1997). Unfortunately, in the present study, chlorophyll was not used as a direct food source for suspension feeders because of a lack of data. Instead, primary production was parameterized with respect to temperature, light, and nutrient limitation (Eqs. 9 and 10). Nevertheless, the results of the input sensitivity tests are consistent with field observations; i.e., there is a harmful effect of reduced freshwater inflow that results in decreased productivity of suspension feeders in both Lavaca and Matagorda Bay (Fig. 15).

Total benthic biomass in Lavaca Bay decreased by 0.4 g dw m⁻² (15%) between changes in salinity that ranged from -50% to +50% of the mean (Figs. 16 a-b). This indicates that reduced inflow rates to this secondary bay would not be beneficial to the benthic community. In contrast, total benthic biomass in Matagorda Bay increased as salinity increased by 5.0 g dw m⁻² (36%) from -50% to +30%, and remained, more or less, constant when changes in salinity ranged +30 and +50% (Fig. 16b). This implies that the benthic system in Matagorda Bay approaches a plateau (or a maximum change) as salinity increases. The beneficial

effect of reduced FWI was clear in Matagorda Bay. It is speculated that the benthos in Matagorda Bay is more adaptable to change in salinity because Matagorda Bay is the primary bay, having direct exchange of sea water. Total benthic biomass in upper San Antonio Bay was found to have similar response to that in Lavaca Bay, revealing that reduced inflow rates caused reduced benthic productivity by 2 g dw m^{-2} (15%) (Fig. 16c), whereas, benthic response from lower San Antonio Bay resembles that in Matagorda Bay showing characters of primary bay. Secondary productivity in this bay increased by 20% (2 g dw m^{-2}) between -50% and +50% changes in salinity (Fig. 16d).

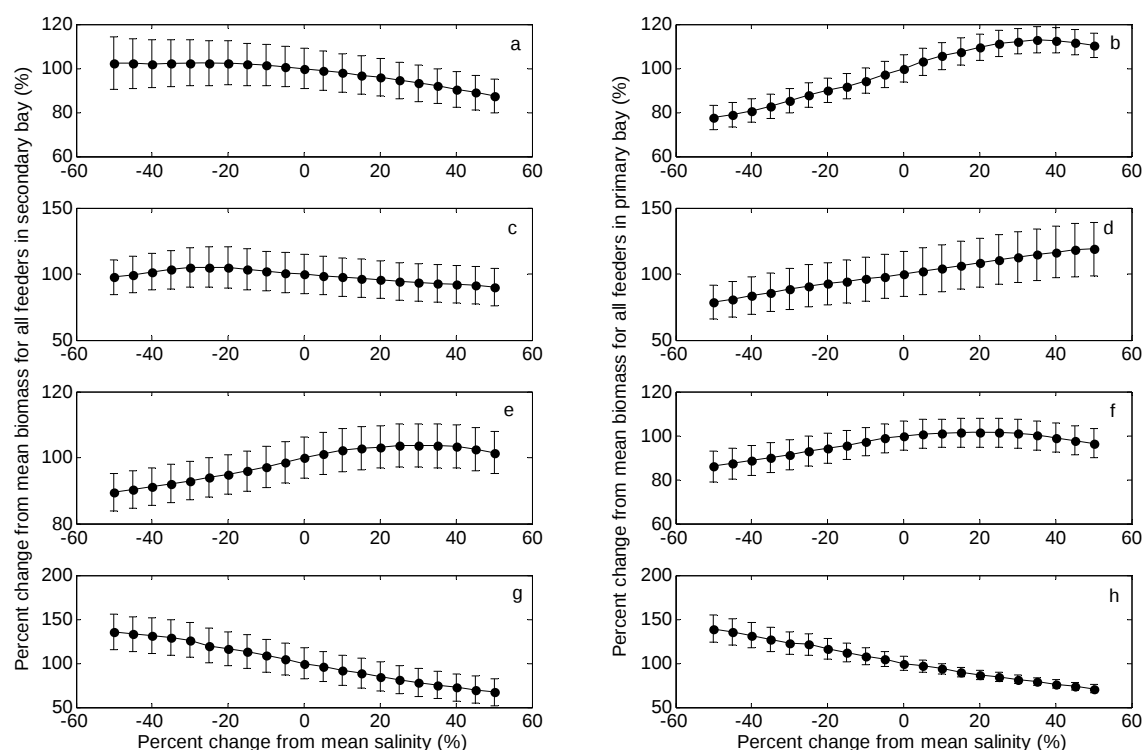


Fig. 16. Percent change from mean benthic biomass for both deposit and suspension feeders simulated with percent change of mean salinity (–50% to 50%) and corresponding nutrients in Lavaca (a); Matagorda (b); upper San Antonio (c); lower San Antonio (d); Nueces (e); Corpus Christi (f); Baffin Bay (g); and upper Laguna Madre (h). The error bars represent the standard error. Long-term mean salinity and nutrient levels found in Table 6.

However, Nueces Bay revealed different benthic responses to reduced freshwater inflow than previous secondary bays. The benthic community in this bay appears to benefit by increased salinity (Fig. 16e). Secondary productivity increased by 14% (0.5 g dw m^{-2}). Corpus Christi Bay followed the same pattern as found in other primary bays with 15% (3 g dw m^{-2}) increase in benthic biomass (Fig. 16f). Interestingly, dramatic responses to changed freshwater inflow occurred at the southern estuarine system. Baffin Bay and upper Laguna Madre revealed 70% (1.3 g dw m^{-2}) and 68% (20.7 g dw m^{-2}) decrease in secondary productivity to reduced FWI, respectively. The upper Laguna Madre did not respond as other primary bays, and this is likely due to seagrass habitat.

Previous studies examined how benthic macrofaunal community structure varied over space and time in response to changes in inflow in Lavaca and Matagorda Bay. Matagorda Bay is a more productive and diverse ecosystem compared to Lavaca Bay (Montagna and Kalke, 1995; Pollack et al., 2009). There is also a direct relationship between freshwater inflow and salinity on benthic communities. Distinct station differences were found in community structure along salinity gradients, which implies that reduced flows will cause upstream communities to take on characteristics of downstream communities. This effect is likely a result of the benthic community acclimating to the elevated salinity by the dominance of more salt tolerant species populating the area. However, there is no direct evidence that the transition in community structure happens in Lavaca Bay.

The simulations of benthic productivity change with changing freshwater inflow (Fig. 16) are based on the assumption that percent changes in salinity correspond with inverse changes in nutrients, based on the relationship between salinity and nutrients (Fig. 2 and Table 2). In general, when salinity increased with decreasing nutrient concentrations, deposit feeder biomass increased while suspension feeder biomass decreased. Although we have not presented the independent effect of each individual limiting factor (i.e., E_{sal} vs. E_{food}), salinity change was designed to have higher sensitivity in model simulations than nutrient change in the structure of governing equations. For example, the effect of nutrient change is included in E_{food} (see Eqs. 6, 9 and 11), which is another limiting effect of E_{ben} (Eq. 3); whereas, the effect of salinity change (Eq. 5) is directly included in Eq. (3). This reflects our best understanding as to how the system is working in the dynamics of the model behavior. In fact, the high sensitivity to salinity revealed in the model was confirmed by observed data. Pollack et al. (2009) used a non-metric multidimensional scaling (nMDS) analysis to illustrate similarities between communities on different sampling periods or between stations. Results of the nMDS analysis showed that distinct benthic

community differences in abundance, biomass, diversity, and species composition are found among the study stations along the salinity gradients.

4.4. Lessons learned

It is difficult to understand benthic dynamics from empirical or static modeling analyses alone (e.g., univariate or multivariate statistical methods). The concept that benthos are an isolated subsystem, governed by internal interactions and “keystone species” is not sufficient to explain the heterogeneity of benthos community dynamics in closely related sites. However, a model can incorporate spatial variability to provide insights into the dynamics and interactions of benthic populations within an ecosystem, or to predict long-term effects of those interactions (Li et al. 1996; Montagna and Li 1997; Kim and Montagna 2009).

Modeling benthic secondary production is challenging because the benthic food web and trophic interactions are complex, and because secondary production rates are not a simple function of physico-chemical variables, as it is with primary production. In this study, we present secondary production in Lavaca and Matagorda Bays simulated by a bioenergetic model. Although the model performance was found to be promising, there are limitations of this heuristic approach because of uncertainties of the governing equations, the large number of fitted parameters (therefore large uncertainty), and the limited number of observed data. Owing to its level of complexity, this modeling exercise is lacking some important mechanistic aspects in explaining the effects of freshwater inflow on benthic ecosystems. Some examples may include: 1) conservative (salinity) or non-conservative properties (nutrients) were forced into the model as surrogates of FWI instead of being simulated within a box (or hydrodynamic) modeling framework; 2) the food sources for deposit and suspension feeders were not dynamically simulated (i.e., ecosystem dynamics for

the lower-trophic level was not included); 3) the 10-cm deep boundary layer (between sediment and water) was arbitrarily chosen; 4) day length was used to represent primary production, which is the food source for suspension feeders instead of a more robust photo-physiological modeling framework (i.e., radiative transfer computation linked to a primary production model); and 5) an uncertainty in the dynamics of benthos and predator interactions.

For instance, Montagna and Kalke (1995) observed how freshwater inflow benefits estuaries in south Texas. *Mulinia lateralis* was one of the dominant mollusks found, and is important as the predominant food source for black drum. *Mulinia* was frequently found in bays along the Texas coast where freshwater inflow has a large influence. It was concluded that recruitment events for *M. lateralis* are likely initiated by a significant change in salinity (Montagna and Kalke 1995). However, with the present approach, we cannot resolve this population dynamical feature because larval recruitment dynamics is not included in the model. Rozas et al. (2005) investigated the effect of freshwater inflow in the Breton Sound Estuary, Louisiana, and found releases of freshwater from the Caernarvon diversion structure led to an increase in submerged aquatic vegetation (SAV) and dissolved oxygen concentrations. Macrofauna populations also increased in density and biomass with increasing inflow, which was probably due to the growing SAV coverage creating more habitats (Rozas et al., 2005). Again, no information is available to incorporate these findings because this modeling study does not include SAV-related habitat for benthic juveniles.

However, all the weaknesses described above are related to the scope of the studies (spatio-temporal resolution and the level of complexity). More elaborate investigations are left for future studies. Nonetheless, using this simple heuristic approach, we were able to learn that the effects of reduced freshwater inflow (and concomitant increases in salinity) have important management implications for water resource development that can influence estuarine

ecosystems. For example, change in benthic community structure, biomass, and diversity occurred when there were changes in the long-term average salinity. This means freshwater inflow plays an important role in maintaining the observed character of estuarine productivity through the combined effects of the frequency, duration, timing, and magnitude of inflow, particularly during droughts or low-flow periods. Interestingly, Texas estuaries are different from one another because the major estuarine systems lie in a climatic gradient where runoff decreases from the Louisiana border in the northeast to the Mexico border in the southwest. Successful application of the model with reduced complexity can be expanded to all Texas estuaries (and possibly to other estuaries) in order to guide development of site-specific estuarine management plans. It is also possible that the presented model will allow predictability of benthic ecosystem responses to future global change (caused by climate and human activities) that may alter the water cycle on Earth.

5. Concluding Remarks

The present study heuristically combined long-term data sets with an energetic model to predict how altering salinity in four Texas estuaries might affect productivity of the macrobenthos population. Based on observed long-term patterns and model predictions, it appears that reducing freshwater inflow may cause components of the benthic community to respond in different ways. Increased salinity benefits deposit feeders, while suspension feeders are harmed. This effect is probably due to the benthic community acclimating to the different salinity regimes, or more (or less) salt tolerant species populating the area. Freshwater inflow into an estuary is recognized as an important factor in estuary productivity, affecting physical, chemical, and biological aspects of the system (Montagna et al., 2002a). River inflow drives increased circulation, salinity gradients, and sediment transport as well as enhancing the productivity of coastal fisheries (Powell et al., 2002). Nutrients from inflow become incorporated into the estuarine food web, can increase vegetation, and enhance secondary production in an area (Rozas et al., 2005). It is clear that inflow is important in maintaining estuarine health and productivity. Management studies should consider not only the quantity of inflow required, but also seek to determine the regime of timing and magnitude of inflow that is needed to maintain functional, healthy ecosystems.

6. References

- Armstrong, N.E., 1985. The ecology of open-bay bottoms of Texas: a community profile. Fish and Wildlife Service. U.S. Department of the Interior, Biological Report 85, 105.
- Brock, D. A., 1994. Estuarine, phytoplankton, primary productivity, and freshwater inflows. In: Longley, W. L. (Ed.), *Freshwater Inflows to Texas Bays and Estuaries: Ecological Relationships and Methods for Determination of Needs*, pp. 74-92.
- Brown, D., and P. Rothery, 1993. *Models in biology: mathematics, statistics and computing*. Oceanography of the Ross Sea, Antarctica. John Wiley & Sons, p. 688.
- Buskey, E.J., P.A. Montagna, A.F. Amos, and T.E. Whitledge, 1997. Disruption of grazer populations as a contributing factor to the initiation of the Texas brown tide algal bloom. *Limnology and Oceanography* 42, 1515-1222.
- Calbet, A., Agusti, S., 1999. Latitudinal changes of copepod egg production rates in Atlantic waters: temperature and food availability as the main driving factors. *Marine Ecology Progress Series* 181, 155-162.
- Carrada, G.C., 1983. Modeling of the Gulf of Naples. In: Carrada, G., Hopkins, T., L, J., Morcos, S. (Eds.), *Quantitative analysis and simulation of Mediterranean coastal ecosystems: The Gulf of Naples, a case study*, UNESCO Reports in Marine Science, pp. 80-153.
- de Hoop, B. J., Herman, P. M. J., Scholten, H., Soetaert, K., 1989. SENECA 1.5: A simulation environment for ecological application. Netherlands Institute of Ecology, Center for Estuarine and Coastal Ecology. 180 pp.
- Fisher, T., Peele, E., Ammerman, J., Harding~Jr, L., 1992. Nutrient limitation of phytoplankton in {Chesapeake Bay}. *Marine Ecology Progress Series* 82, 51-63.
- Friedrichs, M., M. E. Carr, R. Barber, M. Scardi, D. Antoine, R.A. Armstrong, I. Asanuma, M. J. Behrenfeld, E.T. Buitenhuis, F. Chai, J. R. Christian, A. M. Ciotti, S. C. Doney, M. Dowell, J. Dunne, B. Gentili, W. Gregg, N. Hoepffner, J. Ishizaka, T. Kameda, I. Lima, J. Marra, F. Mélin, K. Moore, A. Morel, R. O'Malley, J. O'Reilly, V. S. Saba, M. Schmeltz, T. J. Smyth, J. Tjiputra, K. Waters, T. K. Westberry, A. Winguth, 2009. Assessing the uncertainties of model estimates of primary productivity in the tropical Pacific Ocean. *Journal of Marine Systems* 76, 113-133.
- Houde, E., 1987. Fish early life dynamics and recruitment variability. *American Fisheries Society Symposium* 2, 17-29.

- Houde, E., 1989. Comparative growth, mortality, and energetics of marine fish larvae: temperature and implied latitudinal effects. *Fisheries Bulletin* 87, 471-495.
- Ivlev, V.S., 1961. Experimental ecology of the feeding of fishes. translated from Russian by D. Scott, Yale University Press, New Haven, Conn.
- Jassby, A., Platt, T., 1976. Mathematical formulation of the relationship between photosynthesis and light for phytoplankton. *Limnology and Oceanography* 21, 540.
- Jung, M., G. Le Maire, S. Zaehle, S. Luyssaert, M. Vetter, G. Churkina, P. Ciais, N. Viovy, and M. Reichstein, 2007. Assessing the ability of three land ecosystem models to simulate gross carbon uptake of forests from boreal to Mediterranean climate in Europe. *Biogeosciences* 4, 647-656.
- Kalke, R.D., Montagna, P.A., 1991. The effect of freshwater inflow on macrobenthos in the Lavaca River Delta and Upper Lavaca Bay, Texas. *Contributions in Marine Science*, 49-71.
- Keen, R.E., Spain, J.D., 1992. Computer Simulation in Biology, a Basic Introduction. John Wiley and Sons, New York, p. 516.
- Kim, H., Montagna, P.A., 2009. Implications of Colorado River (Texas, USA) freshwater inflow to benthic ecosystem dynamics: A modeling study. , *Estuarine, Coastal and Shelf Science* 83, 491-504.
- Lotka, A.J., 1925. Elements of physical biology. Williams & Wilkins, Baltimore, Maryland, p. 460.
- Mayzaud, P., Poulet, S.A., 1978. The importance of the time factor in the response of zooplankton to varying concentrations of naturally occurring particulate matter. *Limnology and Oceanography* 23, 1144-1154.
- Montagna, P.A. 2000. Effect of freshwater inflow on macrobenthos productivity and nitrogen losses in Texas estuaries. Final report to Texas Water Development Board, Contract No. 2000-483-323, University of Texas Marine Science Institute Technical Report Number TR/00-03, Port Aransas, Texas. 78 pp.
- Montagna, P.A., M. Alber, P. Doering, and M.S. Connor. 2002a. Freshwater inflow: Science, policy, management. *Estuaries* 25:1243-1245.
- Montagna, P.A., J. Brenner, J. Gibeaut, and S. Morehead. 2009. Coastal impacts. In: *The Impact of Global Warming on Texas*, 2nd Edition. Jurgen Schmandt, Judith Clarkson and Gerald R. North (eds.), University of Texas Press. Chapter 4 (In Press).
- Montagna, P. A., J. C. Gibeaut and J.W. Tunnell Jr. 2007. South Texas Climate 2100: Coastal Impacts. In: J. Norwine and K. John (eds.), *South Texas*

- Climate 2100: Problems and Prospects, Impacts and Implications. . CREST-RESSACA. Texas A&M University-Kingsville, Kingsville, Texas. Chapter 3, pp. 57-77.
- Montagna, P.A., Kalke, R.D., 1992. The effect of freshwater inflow on meiofaunal and macrofaunal populations in the Guadalupe and Nueces estuaries, Texas. *Estuaries*, 307-326.
- Montagna, P.A., Kalke, R.D, 1995. Ecology of infaunal Mollusca in south Texas estuaries. *American Malacological Bulletin* 11, 163-175.
- Montagna, P.A., R.D. Kalke, and C. Ritter, 2002b. Effect of restored freshwater inflow on macrofauna and meiofauna in upper Rincon Bayou, Texas, USA. *Estuaries* 25, 1436-1447.
- Montagna, P.A., Li, J., 1996. Modeling and monitoring long-term change in macrobenthos in Texas estuaries. Final Report to the Texas Water Development Board. University of Texas at Austin, Marine Science Institute, Technical Report No. TR/96-001, Port Aransas, Texas, 149 pp.
- Montagna, P.A., Li, J., 1997. Modeling contaminant effects on deposit feeding nematodes near Gulf of Mexico production platforms. *Ecological Modeling* 98, 151-162.
- Montagna, P.A., Stockwell, A., Kalke, R.D., 1993. Dwarf surfclam *Mulina lateralis* (Say, 1822) populations and feeding during the Texas brown tide event. *Journal of Shellfish Research*, 433-442.
- Morales, P., M. T. Sykes, I. C. Prentice, P. Smith, B. Smith, H. Bugmann, B. Zierl, P. Friedlingstein, N. Viovy, S. Sabaté, A. Sánchez, E. Pla, C. A. Gracia, S. Sitch, A. Arneth, J. Ogee, 2005. Comparing and evaluating process-based ecosystem model predictions of carbon and water fluxes in major European forest biomes. *Global Change Biology* 11, 2211-2233.
- Neale, P., Richerson, P., 1987. Photoinhibition and the diurnal variation of phytoplankton photosynthesis- I. Development of a photosynthesis-irradiance model from studies of *in situ* responses. *Journal of Plankton Research* 9, 167-193.
- Odum, H., 1971. *Environment Power and Society*. John Wiley and Sons, New York.
- Odum, H., 1983. *Systems Ecology* (reprinted as *Ecological and General Systems*). University Press of Colorado, Niwot, Colorado, p. 644.
- O'Neill, R.V., Gardner, R.H., 1979. Sources of uncertainty in ecological model. In: Zeigler, B.P., Elzas, M.S., Klit, G.J., Oren, T.I. (Eds.), *Methodology in system modelling and simulation*. North-Holland Publishing Company, Amsterdam, Netherland, pp. 447-463.

- Palmer, T.A., P.A. Montagna, and R.D. Kalke.,2002. Downstream effects of restored freshwater inflow to Rincon Bayou, Nueces Delta, Texas, USA. *Estuaries* 25, 1448-1456.
- Pepin, P., 1991. Effect of temperature and size on development, mortality, and survival rates of the pelagic early life history stages of marine fish. *Canadian Journal of Fisheries and Aquatic Sciences* 48, 503-518.
- Platt, T., Gallegos, C., Harrison, W., 1980. Photoinhibition of photosynthesis in natural assemblages of marine phytoplankton. *Journal of Marine Research* 38, 687-701.
- Platt, T., Jassby, A., 1976. The relationship between photosynthesis and light for natural assemblages of coastal marine phytoplankton. *Journal of Phycology* 12, 421-430.
- Platt, T., Sathyendranath, S., Caverhill, C., Lewis, M., 1988. Oceanic primary production and available light, further algorithms for remote sensing. *Deep-Sea Research* 35, 855-879.
- Pollack, J.B., J.W. Kinsey, and P.A. Montagna, 2009. Freshwater inflow biotic index (FIBI) for the Lavaca-Colorado estuary, Texas. *Environmental Bioindicators* 4(2), 153-169.
- Poulet, S., Ianora, A., Laabir, M., Klein Breteler, W.C.M., 1995. Towards the measurement of secondary production and recruitment in copepods. *ICES Journal of Marine Science* 52, 359-368.
- Powell, G.L., Matsumoto, J., Brock, D.A., 2002. Methods for determining minimum freshwater inflow needs of Texas bays and estuaries. *Estuaries* 25, 1262-1274.
- Redfield, A.C., 1934. On the proportions of organic derivations in seawater and their relation to the composition of plankton. In: Daniel, R.J. (Ed.), *James Johnson Memorial Volume*. University Press of Liverpool, pp. 177-192.
- Rozas, L.P., Minello, T.J., Munuera-Fernandez, I., Fry, B., Wissel, B., 2005. Macrofaunal distributions and habitat change following winter-spring releases of freshwater into the Breton Sound estuary, Louisiana (USA). *Estuarine Coastal and Shelf Science* 65, 319-336.
- Stockwell, D.A., 1989. Effects of freshwater inflow on the primary production of a Texas coastal bay system. Final Report, Data Synthesis Study (NIPS). The University of Texas Marine Science Institute, Technical Report No. TR/89-010, Port Aransas, Texas.
- Tenore, K.R., Zajac, R.N., Terwin, J., Andrade, J.F.B., Boynton, W., Carey, D., Diaz, R., Holland, A.F., Lopez-Jamar, E., Montagna, P., Nichols, F., Rosenberg, R., Queiroga, H., Sprung, M., Whitlatch, R.B., 2006. Characterizing the role benthos plays in large coastal seas and estuaries: A

modular approach. *Journal of Experimental Marine Biology and Ecology* 330, 392-402.

- Tilman, D., Kilham, S. S., 1976. Phosphate and silicate growth and uptake kinetics of the diatoms *Asterionella formosa* and *Cyclotella meneghiniana* in batch and semicontinuous culture. *Journal of Phycology* 12, 375-383.
- Tolan, J.M. 2007. El Niño-Southern Oscillation impacts translated to the watershed scale: estuarine salinity patterns along the Texas Gulf coast, 1982 to 2004. *Estuarine Coastal and Shelf Science* 72:247-260.
- Walters, C., 1986. Adaptive management of renewable resources. MacMillan Publishing Company, New York.
- Wohlschlag, D.E., Wakeman, J.M., Vetter, R., Ilg, R.G., 1977. Analysis of freshwater inflow effects on metabolic stresses of South Texas bay and estuarine fishes: continuation and extension. Report to Texas Department of Water Resources. The University of Texas Marine Science Institute, Port Aransas, Texas., p. 105.
- Xu, J. and R. Hood, 2006. Modeling biogeochemical cycles in Chesapeake Bay with a coupled physical-biological model. *Estuarine, Coastal and Shelf Science* 69, 19-46.

7. FORTRAN Programs

The SENECA program has automatically created the FORTRAN programs and part of subprograms for the model during the setup of the subprograms for the variances, parameters, forcing functions, calibration methods, and running limitations. The subprograms we created are to describe the all relationships among those variances, parameters and forcing functions. The following are three subprograms we used for the model.

List of Programs in the Appendix:

1. TEXAS.FOR
2. DEPOSIT.FOR
3. EPIGROW.FOR
4. XSIM0.FOR
5. XFORC.FOR
6. XSTART.FOR
7. XCAL.FOR
8. XSENS.FOR

7.1 TEXAS.FOR

```
C#####
C SENECA 2.0 (C) NI00-CEM0/DGW
C File: "model".FOR
C Date: 1-12-92
C Version: 4
C This file contains the main program for the single run program of
C the model.
C#####
      PROGRAM TEXAS
C
C-----
C Declarations:
C
      INTEGER ERRCODE, ERRIND, TLCODE, TYPRUN, TUNIT, TZERO,
      & FU, FURPR, FUSTR, FUVAR, FUPAR, FUBND, FUWST, FUFRC, FURAN, FUDAT, FURES,
      & FUTLG, FULOG,
      & FRSRES, FRSDAT, FPIRES, FPRRES, FNVRES, FRSRAN, FPTRES, FPTRAN, LOGMODE,
      & NSVAR, NBOUND, NWASTE, NFORC, NOUTVAR, NRANPAR, NVALVAR, MVALVAR,
      & NTLVAR, IOUTSTEP, IDRLABCH
      REAL DELT, DELTA, MODTIME, MXABSCG, MNCHANGE, MNSWITCH, MNABSCG,
      & MXRLABCH, NOVALUE, MAXVALUE
      COMMON /XCBSIM/ ERRCODE, ERRIND, TLCODE, TYPRUN, TUNIT, TZERO,
      & FU, FURPR, FUSTR, FUVAR, FUPAR, FUBND, FUWST, FUFRC, FURAN, FUDAT, FURES,
      & FUTLG, FULOG,
      & FRSRES, FRSDAT, FPIRES, FPRRES, FNVRES, FRSRAN, FPTRES, FPTRAN, LOGMODE,
      & NSVAR, NBOUND, NWASTE, NFORC, NOUTVAR, NRANPAR, NVALVAR, MVALVAR,
      & NTLVAR, IOUTSTEP, IDRLABCH,
      & DELT, DELTA, MODTIME, MXABSCG, MNCHANGE, MNSWITCH, MNABSCG,
      & MXRLABCH, NOVALUE, MAXVALUE
C
      INTEGER XMXGOF
      PARAMETER (XMXGOF = 8)
      INTEGER TSTART(0:6), RPI02, TSTOP(0:6), ITEND,
      & OUTFACT, NOUTSTEP, MXSTEPS, RPI05, GOFRES, GOFVAR, MNVALSTEP, NBAND,
      & IRUN, NRANRUN, NCALRUN, IBAND, NCENTR, ISEED, IBCONT, STOPCODE,
      & FIDENT, INTEGRATE, TIMEOFF, RPI09, IBACTPAR, RPI10(2:4)
      REAL
      MXDELT, MNDELT, MXCHANGE, MXSWITCH, MNRELCHG, RPR02(2:4), RPR03(4),
      & GOFERR(XMXGOF)
      COMMON /XCBRPR/ TSTART, RPI02, TSTOP, ITEND,
      & OUTFACT, NOUTSTEP, MXSTEPS, RPI05, GOFRES, GOFVAR, MNVALSTEP, NBAND,
      & IRUN, NRANRUN, NCALRUN, IBAND, NCENTR, ISEED, IBCONT, STOPCODE,
      & FIDENT, INTEGRATE, TIMEOFF, RPI09, IBACTPAR, RPI10,
      & MXDELT, MNDELT, MXCHANGE, MXSWITCH, MNRELCHG, RPR02, RPR03, GOFERR
C
      INTEGER TYPSIM, TYPSENS, TYPCAL
      PARAMETER (TYPSIM = 1)
      PARAMETER (TYPSENS = 2)
      PARAMETER (TYPCAL = 3)
C-----
C Statements:
C
C Initialize single run simulation and read run parameters
```

```

        CALL XINSIM(TYPSIM)
        IF (ERRCODE .NE. 0) GOTO 100
C
C Initialize variables, parameters and forcing functions
        CALL XRDSTRUCT
        IF (ERRCODE .NE. 0) GOTO 100
C
C Create result file
        CALL XOPRESULT
C
C Do a single run simulation
        CALL XSIMULATE
C
C Write results simulation to result file
        CALL XWRRESULT
C
C Stop simulation
        IF (ERRCODE .EQ. 0) STOPCODE = 1
100    CALL XSTOP
C
        END
C End of program.
C
C#### End of File #####

```

7.2 DEPOSIT.FOR

```
C#####
C SENECA 1.85 (C) N100-CEM0/DGW
C Date: 1-7-92
C#####

      SUBROUTINE DEPOSIT(TIME)
C
      IMPLICIT REAL(A-Z)
C Parameter:
      REAL TIME
C
C Submodel routine
C-----
C Declarations:

      INCLUDE 'TEXAS.DCS'
      INCLUDE 'DEPOSIT.DCP'
      INCLUDE 'DEPOSIT.DCV'
      INCLUDE 'XSIM0.DEX'
C
C-----
C Statements:
C

      integer j

      real c(8)
      real dep(8),ddep(8),epi(8)
      real aledep(8)
      real temp(8),mic(8),sal(8),scale
      real pred1(8),pred2(8),pred3(8)

      equivalence(dep(1),dep1)
      equivalence(dep(2),dep2)
      equivalence(dep(3),dep3)
      equivalence(dep(4),dep4)
      equivalence(dep(5),dep5)
      equivalence(dep(6),dep6)
      equivalence(dep(7),dep7)
      equivalence(dep(8),dep8)

      equivalence(epi(1),epi1)
      equivalence(epi(2),epi2)
      equivalence(epi(3),epi3)
      equivalence(epi(4),epi4)
      equivalence(epi(5),epi5)
      equivalence(epi(6),epi6)
      equivalence(epi(7),epi7)
      equivalence(epi(8),epi8)

      equivalence(mic(1),mic1)
```

```

equivalence(mic(2),mic2)
equivalence(mic(3),mic3)
equivalence(mic(4),mic4)
equivalence(mic(5),mic5)
equivalence(mic(6),mic6)
equivalence(mic(7),mic7)
equivalence(mic(8),mic8)

equivalence(ddep(1),ddep1)
equivalence(ddep(2),ddep2)
equivalence(ddep(3),ddep3)
equivalence(ddep(4),ddep4)
equivalence(ddep(5),ddep5)
equivalence(ddep(6),ddep6)
equivalence(ddep(7),ddep7)
equivalence(ddep(8),ddep8)

equivalence(c(1),c1)
equivalence(c(2),c2)
equivalence(c(3),c3)
equivalence(c(4),c4)
equivalence(c(5),c5)
equivalence(c(6),c6)
equivalence(c(7),c7)
equivalence(c(8),c8)

sal(1)=sal1(time)
sal(2)=sal2(time)
sal(3)=sal3(time)
sal(4)=sal4(time)
sal(5)=sal5(time)
sal(6)=sal6(time)
sal(7)=sal7(time)
sal(8)=sal8(time)

temp(1)=temp1(time)
temp(2)=temp2(time)
temp(3)=temp3(time)
temp(4)=temp4(time)
temp(5)=temp5(time)
temp(6)=temp6(time)
temp(7)=temp7(time)
temp(8)=temp8(time)

pred1(1)=pred11(time)
pred1(2)=pred12(time)
pred1(3)=pred13(time)
pred1(4)=pred14(time)
pred1(5)=pred15(time)
pred1(6)=pred16(time)
pred1(7)=pred17(time)
pred1(8)=pred18(time)

pred2(1)=pred21(time)
pred2(2)=pred22(time)
pred2(3)=pred23(time)
pred2(4)=pred24(time)

```

```

pred2(5)=pred25(time)
pred2(6)=pred26(time)
pred2(7)=pred27(time)
pred2(8)=pred28(time)

```

```

pred3(1)=pred31(time)
pred3(2)=pred32(time)
pred3(3)=pred33(time)
pred3(4)=pred34(time)
pred3(5)=pred35(time)
pred3(6)=pred36(time)
pred3(7)=pred37(time)
pred3(8)=pred38(time)

```

C-----
C Statements:

```

c j=1-8 for 8 bays
c dep(j) for deposit-feeders
c epi(j) for microfauna-feeders

```

```

      j=bay

```

```

c***** cut off unused parameters *****

```

```

c1      do 5 j=bay,bay+1
        if (dep(j).gt.1.e30)then
c          dep(j)=xdiv(1,0.)          !H-C KIM(2009)
          CALL XERROR(20)             !H-C KIM(2009) to avoid
floating-point error
        else
          if (dep(j).lt.1.e-30)then
c          dep(j)=xdiv(1,0.)          !H-C KIM(2009)
          CALL XERROR(20)             !H-C KIM(2009) to avoid
floating-point error
        else
          endif
        endif
c5      continue

```

```

c      do 10 j=bay,bay+1

```

```

c => c(j)=organic carbon % in the sediment, few observed data are
available

```

```

c      c(j)=poc(j)/(pm0*0.42*100*100*10)*100 ! By Jian Li
c(j)=poc(j)/(pm0/34.5*100*100*10)*100 ! By Hae-Cheol Kim
(2008)

```

```

c***** production *****
c Combine intake rate,assimilation efficiency and respiration rate
c     together,
c a Lotka-Volterra logistic model plus temperature and salinity effects
c     for limited population growth is used here for biomass growth:
c [monthly growth rate] = [annual P/B]/12 = p8/12

c***** temperature and salinity effects *****

        scale=1.00+0.0    ! for sensitivity

c longterm average for sensitivity
c     sal(1)=15.28
c     sal(2)=24.67
c     temp(1)=21.49
c     temp(2)=21.79
c-----

c ORIGINAL
c     teffdep(j)=1/exp(abs(temp(j)-31.5)/pm1)
c KIM2008
c     teffdep(j)=1/exp(abs(temp(j)-pm10a)/(pm1))
c     teffdep(j)=1.-exp(-(temp(j)-0.7)/40.)

        seffdep(j)=1/exp(abs(sal(j)*scale-pm3A)/(pm2))

c     feffdep(j)=xdiv(xdiv(poc(j),dep(j)),xdiv(poc(j),dep(j))+pm4a)
!Jian Li (1996)

C     feffdep(j)=xdiv(xdiv(poc(j),1.),xdiv(poc(j),1.))+pm4a) ! Hae-
Cheol Kim (2008) This is more correct but has NO big difference
        feffdep(j)=poc(j)/(poc(j)+pm4a) ! Hae-Cheol Kim (2009) Same as
above but with different expression

        aledep(j)=teffdep(j)*seffdep(j)*feffdep(j)

        prodep(j)=dep(j)*pm8A/12*(1-dep(j)/(pm9A))
&          *aledep(j)

        prrdep(j)=prodep(j)/dep(j)*12

c***** add the predation effect *****

        ddep(j)=ddep(j)+prodep(j)

c ORIGINAL (by Jian Li)
c     byodep(j)=30.*
c     & (1-(exp(-pm7*dep(j))))*

```

```

c      & (pm61a*pred1(j)+pm62a*pred2(j)+pm63a*pred3(j))
cc      & (1-(exp(-3.77488E-3*dep(j))))*
cc      & (1.228183*pred3(j))
c MAPS et al., 2005 (Ivlev's type eq. used in MEPS 285:117-128)
c      byodep(j)=30.*
c      & 2000*(1.-EXP(-0.01*(temp(j)-0.2)))*
c      & pm7a*(1.-EXP(-pm7a*dep(j)))*
c      & (pred1(j)+pred2(j)+pred3(j))*dep(j)
c KIM2008 (Ivlev's eq. used with parameterized 10**) - best working
c      byodep(j)=30.*
c      & 10**(pm7a2*(temp(j)-pm7a3))*
c      & pm7a*(1.-EXP(-pm7a*dep(j)))* ! Ivlev's
type
c      & (pred1(j)+pred2(j)+pred3(j))*dep(j)
c KIM2008 (Ivlev's eq. used with fixed 10**)
c      byodep(j)=30.*
c      & 1.44*10**(0.0275*(temp(j)-0.7))*
cc      & 0.05*(1.-EXP(-0.05*dep(j)))* ! good-
working (replaced pm7a below with 0.05)
c      & pm7a*(1.-EXP(-pm7a*dep(j)))* ! Ivlev's
type
c      & (pred1(j)+pred2(j)+pred3(j))*dep(j)

```

ddep(j) = ddep(j)-byodep(j)

c10 continue

END

C*****

7.3 EPIGROW.FOR

```
C#####
C SENECA 1.85 (C) NI00-CEM0/DGW
C Date: 1-7-92
C#####
```

```
      SUBROUTINE DEPOSIT(TIME)
C
      IMPLICIT REAL(A-Z)
C Parameter:
      REAL TIME
C
C Submodel routine
C-----
C Declarations:
      INCLUDE 'TEXAS.DCS'
      INCLUDE 'DEPOSIT.DCP'
      INCLUDE 'DEPOSIT.DCV'
      INCLUDE 'XSIM0.DEX'
C
C-----
C Statements:
C
      integer j

      real c(8)
      real dep(8),ddep(8),epi(8)
      real aledep(8)
      real temp(8),mic(8),sal(8),scale
      real pred1(8),pred2(8),pred3(8)

      equivalence(dep(1),dep1)
      equivalence(dep(2),dep2)
      equivalence(dep(3),dep3)
      equivalence(dep(4),dep4)
      equivalence(dep(5),dep5)
      equivalence(dep(6),dep6)
      equivalence(dep(7),dep7)
      equivalence(dep(8),dep8)

      equivalence(epi(1),epi1)
      equivalence(epi(2),epi2)
      equivalence(epi(3),epi3)
      equivalence(epi(4),epi4)
      equivalence(epi(5),epi5)
      equivalence(epi(6),epi6)
      equivalence(epi(7),epi7)
      equivalence(epi(8),epi8)

      equivalence(mic(1),mic1)
```

```

equivalence(mic(2),mic2)
equivalence(mic(3),mic3)
equivalence(mic(4),mic4)
equivalence(mic(5),mic5)
equivalence(mic(6),mic6)
equivalence(mic(7),mic7)
equivalence(mic(8),mic8)

equivalence(ddep(1),ddep1)
equivalence(ddep(2),ddep2)
equivalence(ddep(3),ddep3)
equivalence(ddep(4),ddep4)
equivalence(ddep(5),ddep5)
equivalence(ddep(6),ddep6)
equivalence(ddep(7),ddep7)
equivalence(ddep(8),ddep8)

equivalence(c(1),c1)
equivalence(c(2),c2)
equivalence(c(3),c3)
equivalence(c(4),c4)
equivalence(c(5),c5)
equivalence(c(6),c6)
equivalence(c(7),c7)
equivalence(c(8),c8)

sal(1)=sal1(time)
sal(2)=sal2(time)
sal(3)=sal3(time)
sal(4)=sal4(time)
sal(5)=sal5(time)
sal(6)=sal6(time)
sal(7)=sal7(time)
sal(8)=sal8(time)

temp(1)=temp1(time)
temp(2)=temp2(time)
temp(3)=temp3(time)
temp(4)=temp4(time)
temp(5)=temp5(time)
temp(6)=temp6(time)
temp(7)=temp7(time)
temp(8)=temp8(time)

pred1(1)=pred11(time)
pred1(2)=pred12(time)
pred1(3)=pred13(time)
pred1(4)=pred14(time)
pred1(5)=pred15(time)
pred1(6)=pred16(time)
pred1(7)=pred17(time)
pred1(8)=pred18(time)

pred2(1)=pred21(time)
pred2(2)=pred22(time)
pred2(3)=pred23(time)
pred2(4)=pred24(time)

```

```

pred2(5)=pred25(time)
pred2(6)=pred26(time)
pred2(7)=pred27(time)
pred2(8)=pred28(time)

```

```

pred3(1)=pred31(time)
pred3(2)=pred32(time)
pred3(3)=pred33(time)
pred3(4)=pred34(time)
pred3(5)=pred35(time)
pred3(6)=pred36(time)
pred3(7)=pred37(time)
pred3(8)=pred38(time)

```

C-----
C Statements:

```

c j=1-8 for 8 bays
c dep(j) for deposit-feeders
c epi(j) for microfauna-feeders

```

```

      j=bay

```

```

c***** cut off unused parameters *****

```

```

c1      do 5 j=bay,bay+1
        if (dep(j).gt.1.e30)then
c          dep(j)=xdiv(1,0.)          !H-C KIM(2009)
          CALL XERROR(20)             !H-C KIM(2009) to avoid
floating-point error
        else
          if (dep(j).lt.1.e-30)then
c          dep(j)=xdiv(1,0.)          !H-C KIM(2009)
          CALL XERROR(20)             !H-C KIM(2009) to avoid
floating-point error
        else
          endif
        endif
c5      continue

```

```

c      do 10 j=bay,bay+1

```

```

c => c(j)=organic carbon % in the sediment, few observed data are
available

```

```

c      c(j)=poc(j)/(pm0*0.42*100*100*10)*100 ! By Jian Li
c      c(j)=poc(j)/(pm0/34.5*100*100*10)*100 ! By Hae-Cheol Kim
(2008)

```

```

c***** production *****
c Combine intake rate,assimilation efficiency and respiration rate
c     together,
c a Lotka-Volterra logistic model plus temperature and salinity effects
c     for limited population growth is used here for biomass growth:
c [monthly growth rate] = [annual P/B]/12 = p8/12

c***** temperature and salinity effects *****

        scale=1.00+0.0    ! for sensitivity

c longterm average for sensitivity
c     sal(1)=15.28
c     sal(2)=24.67
c     temp(1)=21.49
c     temp(2)=21.79
c-----

c ORIGINAL
c     teffdep(j)=1/exp(abs(temp(j)-31.5)/pm1)
c KIM2008
c     teffdep(j)=1/exp(abs(temp(j)-pm10a)/(pm1))
c     teffdep(j)=1.-exp(-(temp(j)-0.7)/40.)

        seffdep(j)=1/exp(abs(sal(j)*scale-pm3A)/(pm2))

c     feffdep(j)=xdiv(xdiv(poc(j),dep(j)),xdiv(poc(j),dep(j))+pm4a)
!Jian Li (1996)

C     feffdep(j)=xdiv(xdiv(poc(j),1.),xdiv(poc(j),1.))+pm4a) ! Hae-
Cheol Kim (2008) This is more correct but has NO big difference
        feffdep(j)=poc(j)/(poc(j)+pm4a) ! Hae-Cheol Kim (2009) Same as
above but with different expression

        aledep(j)=teffdep(j)*seffdep(j)*feffdep(j)

        prodep(j)=dep(j)*pm8A/12*(1-dep(j)/(pm9A))
&          *aledep(j)

        prrdep(j)=prodep(j)/dep(j)*12

c***** add the predation effect *****

        ddep(j)=ddep(j)+prodep(j)

c ORIGINAL (by Jian Li)
c     byodep(j)=30.*
c     & (1-(exp(-pm7*dep(j))))*

```

```

c      & (pm61a*pred1(j)+pm62a*pred2(j)+pm63a*pred3(j))
cc      & (1-(exp(-3.77488E-3*dep(j))))*
cc      & (1.228183*pred3(j))
c MAPS et al., 2005 (Ivlev's type eq. used in MEPS 285:117-128)
c      byodep(j)=30.*
c      & 2000*(1.-EXP(-0.01*(temp(j)-0.2)))*
c      & pm7a*(1.-EXP(-pm7a*dep(j)))*
c      & (pred1(j)+pred2(j)+pred3(j))*dep(j)
c KIM2008 (Ivlev's eq. used with parameterized 10**) - best working
c      byodep(j)=30.*
c      & 10**(pm7a2*(temp(j)-pm7a3))*
c      & pm7a*(1.-EXP(-pm7a*dep(j)))* ! Ivlev's type
c      & (pred1(j)+pred2(j)+pred3(j))*dep(j)
c KIM2008 (Ivlev's eq. used with fixed 10**)
c      byodep(j)=30.*
c      & 1.44*10**(0.0275*(temp(j)-0.7))*
cc      & 0.05*(1.-EXP(-0.05*dep(j)))* ! good-
working (replaced pm7a below with 0.05)
c      & pm7a*(1.-EXP(-pm7a*dep(j)))* ! Ivlev's type
c      & (pred1(j)+pred2(j)+pred3(j))*dep(j)

```

ddep(j) = ddep(j)-byodep(j)

c10 continue

END

C*****

7.4 XSIM0.FOR

```
C#####
C SENECA 2.0 (C) NI00-CEM0/DGW
C File: XSIM0.FOR
C Model: TEXAS
C Creation date: 30-1-2008
C This file contains the routine that calls all submodel routines
(XSUBMODS)
C#####
C
C      SUBROUTINE XSUBMODS(TIME)
C      REAL TIME
C
C      CALL DEPOSIT(TIME)
C      CALL EPIGROW(TIME)
C      RETURN
C      END
C of XSUBMODS
C
C      INCLUDE 'XFORC.FOR'
C      INCLUDE 'XBOUND.FOR'
C      INCLUDE 'XWASTE.FOR'
C      INCLUDE 'XTLAG.FOR'
```

7.5 XFORC.FOR

```
C#####
C SENECA 2.0 (C) NI00-CEM0/DGW
C File: XFORC.FOR
C Model: TEXAS
C Creation date: 30-1-2008
C This file contains all Forcing functions declarations.
C#####
C
C      REAL FUNCTION PRED11(TIME)
C      REAL TIME
C      EXTERNAL XTIMSER
C      PRED11 = XTIMSER(3,1,0,TIME)
C      RETURN
C      END
C of PRED11
C
C      REAL FUNCTION PRED21(TIME)
C      REAL TIME
C      EXTERNAL XTIMSER
C      PRED21 = XTIMSER(3,2,0,TIME)
C      RETURN
C      END
C of PRED21
C
C      REAL FUNCTION PRED31(TIME)
C      REAL TIME
C      EXTERNAL XTIMSER
C      PRED31 = XTIMSER(3,3,0,TIME)
C      RETURN
C      END
C of PRED31
C
C      REAL FUNCTION PRED12(TIME)
C      REAL TIME
C      EXTERNAL XTIMSER
C      PRED12 = XTIMSER(3,4,0,TIME)
C      RETURN
C      END
C of PRED12
C
C      REAL FUNCTION PRED22(TIME)
C      REAL TIME
C      EXTERNAL XTIMSER
C      PRED22 = XTIMSER(3,5,0,TIME)
C      RETURN
C      END
C of PRED22
C
C      REAL FUNCTION PRED32(TIME)
C      REAL TIME
C      EXTERNAL XTIMSER
C      PRED32 = XTIMSER(3,6,0,TIME)
C      RETURN
```

```

        END
C of PRED32
C
    REAL FUNCTION PRED13(TIME)
    REAL TIME
    EXTERNAL XTIMSER
    PRED13 = XTIMSER(3,7,0,TIME)
    RETURN
    END
C of PRED13
C
    REAL FUNCTION PRED23(TIME)
    REAL TIME
    EXTERNAL XTIMSER
    PRED23 = XTIMSER(3,8,0,TIME)
    RETURN
    END
C of PRED23
C
    REAL FUNCTION PRED33(TIME)
    REAL TIME
    EXTERNAL XTIMSER
    PRED33 = XTIMSER(3,9,0,TIME)
    RETURN
    END
C of PRED33
C
    REAL FUNCTION PRED14(TIME)
    REAL TIME
    EXTERNAL XTIMSER
    PRED14 = XTIMSER(3,10,0,TIME)
    RETURN
    END
C of PRED14
C
    REAL FUNCTION PRED24(TIME)
    REAL TIME
    EXTERNAL XTIMSER
    PRED24 = XTIMSER(3,11,0,TIME)
    RETURN
    END
C of PRED24
C
    REAL FUNCTION PRED34(TIME)
    REAL TIME
    EXTERNAL XTIMSER
    PRED34 = XTIMSER(3,12,0,TIME)
    RETURN
    END
C of PRED34
C
    REAL FUNCTION PRED15(TIME)
    REAL TIME
    EXTERNAL XTIMSER
    PRED15 = XTIMSER(3,13,0,TIME)
    RETURN
    END

```

```

C of PRED15
C
    REAL FUNCTION PRED25(TIME)
    REAL TIME
    EXTERNAL XTIMSER
    PRED25 = XTIMSER(3,14,0,TIME)
    RETURN
    END
C of PRED25
C
    REAL FUNCTION PRED35(TIME)
    REAL TIME
    EXTERNAL XTIMSER
    PRED35 = XTIMSER(3,15,0,TIME)
    RETURN
    END
C of PRED35
C
    REAL FUNCTION PRED16(TIME)
    REAL TIME
    EXTERNAL XTIMSER
    PRED16 = XTIMSER(3,16,0,TIME)
    RETURN
    END
C of PRED16
C
    REAL FUNCTION PRED26(TIME)
    REAL TIME
    EXTERNAL XTIMSER
    PRED26 = XTIMSER(3,17,0,TIME)
    RETURN
    END
C of PRED26
C
    REAL FUNCTION PRED36(TIME)
    REAL TIME
    EXTERNAL XTIMSER
    PRED36 = XTIMSER(3,18,0,TIME)
    RETURN
    END
C of PRED36
C
    REAL FUNCTION PRED17(TIME)
    REAL TIME
    EXTERNAL XTIMSER
    PRED17 = XTIMSER(3,19,0,TIME)
    RETURN
    END
C of PRED17
C
    REAL FUNCTION PRED27(TIME)
    REAL TIME
    EXTERNAL XTIMSER
    PRED27 = XTIMSER(3,20,0,TIME)
    RETURN
    END
C of PRED27

```

```

C      REAL FUNCTION PRED37(TIME)
        REAL TIME
        EXTERNAL XTIMSER
        PRED37 = XTIMSER(3, 21, 0, TIME)
        RETURN
      END
C of PRED37
C      REAL FUNCTION PRED18(TIME)
        REAL TIME
        EXTERNAL XTIMSER
        PRED18 = XTIMSER(3, 22, 0, TIME)
        RETURN
      END
C of PRED18
C      REAL FUNCTION PRED28(TIME)
        REAL TIME
        EXTERNAL XTIMSER
        PRED28 = XTIMSER(3, 23, 0, TIME)
        RETURN
      END
C of PRED28
C      REAL FUNCTION PRED38(TIME)
        REAL TIME
        EXTERNAL XTIMSER
        PRED38 = XTIMSER(3, 24, 0, TIME)
        RETURN
      END
C of PRED38
C      REAL FUNCTION DAYL(TIME)
        REAL TIME
        EXTERNAL XTIMSER
        DAYL = XTIMSER(3, 25, 0, TIME)
        RETURN
      END
C of DAYL
C      REAL FUNCTION TEMP1(TIME)
        REAL TIME
        EXTERNAL XTIMSER
        TEMP1 = XTIMSER(3, 26, 0, TIME)
        RETURN
      END
C of TEMP1
C      REAL FUNCTION TEMP2(TIME)
        REAL TIME
        EXTERNAL XTIMSER
        TEMP2 = XTIMSER(3, 27, 0, TIME)
        RETURN
      END
C of TEMP2
C

```

```

      REAL FUNCTION TEMP3(TIME)
      REAL TIME
      EXTERNAL XTIMSER
      TEMP3 = XTIMSER(3,28,0,TIME)
      RETURN
      END
C of TEMP3
C
      REAL FUNCTION TEMP4(TIME)
      REAL TIME
      EXTERNAL XTIMSER
      TEMP4 = XTIMSER(3,29,0,TIME)
      RETURN
      END
C of TEMP4
C
      REAL FUNCTION TEMP5(TIME)
      REAL TIME
      EXTERNAL XTIMSER
      TEMP5 = XTIMSER(3,30,0,TIME)
      RETURN
      END
C of TEMP5
C
      REAL FUNCTION TEMP6(TIME)
      REAL TIME
      EXTERNAL XTIMSER
      TEMP6 = XTIMSER(3,31,0,TIME)
      RETURN
      END
C of TEMP6
C
      REAL FUNCTION TEMP7(TIME)
      REAL TIME
      EXTERNAL XTIMSER
      TEMP7 = XTIMSER(3,32,0,TIME)
      RETURN
      END
C of TEMP7
C
      REAL FUNCTION TEMP8(TIME)
      REAL TIME
      EXTERNAL XTIMSER
      TEMP8 = XTIMSER(3,33,0,TIME)
      RETURN
      END
C of TEMP8
C
      REAL FUNCTION DEPTH1(TIME)
      REAL TIME
      EXTERNAL XTIMSER
      DEPTH1 = XTIMSER(3,34,0,TIME)
      RETURN
      END
C of DEPTH1
C
      REAL FUNCTION DEPTH2(TIME)

```

```

        REAL TIME
        EXTERNAL XTIMSER
        DEPTH2 = XTIMSER(3, 35, 0, TIME)
        RETURN
    END
C of DEPTH2
C
    REAL FUNCTION DEPTH3(TIME)
    REAL TIME
    EXTERNAL XTIMSER
    DEPTH3 = XTIMSER(3, 36, 0, TIME)
    RETURN
    END
C of DEPTH3
C
    REAL FUNCTION DEPTH4(TIME)
    REAL TIME
    EXTERNAL XTIMSER
    DEPTH4 = XTIMSER(3, 37, 0, TIME)
    RETURN
    END
C of DEPTH4
C
    REAL FUNCTION DEPTH5(TIME)
    REAL TIME
    EXTERNAL XTIMSER
    DEPTH5 = XTIMSER(3, 38, 0, TIME)
    RETURN
    END
C of DEPTH5
C
    REAL FUNCTION DEPTH6(TIME)
    REAL TIME
    EXTERNAL XTIMSER
    DEPTH6 = XTIMSER(3, 39, 0, TIME)
    RETURN
    END
C of DEPTH6
C
    REAL FUNCTION DEPTH7(TIME)
    REAL TIME
    EXTERNAL XTIMSER
    DEPTH7 = XTIMSER(3, 40, 0, TIME)
    RETURN
    END
C of DEPTH7
C
    REAL FUNCTION DEPTH8(TIME)
    REAL TIME
    EXTERNAL XTIMSER
    DEPTH8 = XTIMSER(3, 41, 0, TIME)
    RETURN
    END
C of DEPTH8
C
    REAL FUNCTION SAL1(TIME)
    REAL TIME

```

```

EXTERNAL XTIMSER
SAL1 = XTIMSER(3,42,0,TIME)
RETURN
END
C of SAL1
C
REAL FUNCTION SAL2(TIME)
REAL TIME
EXTERNAL XTIMSER
SAL2 = XTIMSER(3,43,0,TIME)
RETURN
END
C of SAL2
C
REAL FUNCTION SAL3(TIME)
REAL TIME
EXTERNAL XTIMSER
SAL3 = XTIMSER(3,44,0,TIME)
RETURN
END
C of SAL3
C
REAL FUNCTION SAL4(TIME)
REAL TIME
EXTERNAL XTIMSER
SAL4 = XTIMSER(3,45,0,TIME)
RETURN
END
C of SAL4
C
REAL FUNCTION SAL5(TIME)
REAL TIME
EXTERNAL XTIMSER
SAL5 = XTIMSER(3,46,0,TIME)
RETURN
END
C of SAL5
C
REAL FUNCTION SAL6(TIME)
REAL TIME
EXTERNAL XTIMSER
SAL6 = XTIMSER(3,47,0,TIME)
RETURN
END
C of SAL6
C
REAL FUNCTION SAL7(TIME)
REAL TIME
EXTERNAL XTIMSER
SAL7 = XTIMSER(3,48,0,TIME)
RETURN
END
C of SAL7
C
REAL FUNCTION SAL8(TIME)
REAL TIME
EXTERNAL XTIMSER

```

```

        SAL8 = XTIMSER(3,49,0,TIME)
        RETURN
    END
C of SAL8
C
    REAL FUNCTION P1(TIME)
    REAL TIME
    EXTERNAL XTIMSER
    P1 = XTIMSER(3,50,0,TIME)
    RETURN
    END
C of P1
C
    REAL FUNCTION P2(TIME)
    REAL TIME
    EXTERNAL XTIMSER
    P2 = XTIMSER(3,51,0,TIME)
    RETURN
    END
C of P2
C
    REAL FUNCTION P3(TIME)
    REAL TIME
    EXTERNAL XTIMSER
    P3 = XTIMSER(3,52,0,TIME)
    RETURN
    END
C of P3
C
    REAL FUNCTION P4(TIME)
    REAL TIME
    EXTERNAL XTIMSER
    P4 = XTIMSER(3,53,0,TIME)
    RETURN
    END
C of P4
C
    REAL FUNCTION P5(TIME)
    REAL TIME
    EXTERNAL XTIMSER
    P5 = XTIMSER(3,54,0,TIME)
    RETURN
    END
C of P5
C
    REAL FUNCTION P6(TIME)
    REAL TIME
    EXTERNAL XTIMSER
    P6 = XTIMSER(3,55,0,TIME)
    RETURN
    END
C of P6
C
    REAL FUNCTION P7(TIME)
    REAL TIME
    EXTERNAL XTIMSER
    P7 = XTIMSER(3,56,0,TIME)

```

```

        RETURN
    END
C of P7
C
    REAL FUNCTION P8(TIME)
    REAL TIME
    EXTERNAL XTIMSER
    P8 = XTIMSER(3,57,0,TIME)
    RETURN
    END
C of P8
C
    REAL FUNCTION N1(TIME)
    REAL TIME
    EXTERNAL XTIMSER
    N1 = XTIMSER(3,58,0,TIME)
    RETURN
    END
C of N1
C
    REAL FUNCTION N2(TIME)
    REAL TIME
    EXTERNAL XTIMSER
    N2 = XTIMSER(3,59,0,TIME)
    RETURN
    END
C of N2
C
    REAL FUNCTION N3(TIME)
    REAL TIME
    EXTERNAL XTIMSER
    N3 = XTIMSER(3,60,0,TIME)
    RETURN
    END
C of N3
C
    REAL FUNCTION N4(TIME)
    REAL TIME
    EXTERNAL XTIMSER
    N4 = XTIMSER(3,61,0,TIME)
    RETURN
    END
C of N4
C
    REAL FUNCTION N5(TIME)
    REAL TIME
    EXTERNAL XTIMSER
    N5 = XTIMSER(3,62,0,TIME)
    RETURN
    END
C of N5
C
    REAL FUNCTION N6(TIME)
    REAL TIME
    EXTERNAL XTIMSER
    N6 = XTIMSER(3,63,0,TIME)
    RETURN

```

```

        END
C of N6
C
    REAL FUNCTION N7(TIME)
    REAL TIME
    EXTERNAL XTIMSER
    N7 = XTIMSER(3,64,0,TIME)
    RETURN
    END
C of N7
C
    REAL FUNCTION N8(TIME)
    REAL TIME
    EXTERNAL XTIMSER
    N8 = XTIMSER(3,65,0,TIME)
    RETURN
    END
C of N8
C
    REAL FUNCTION SI1(TIME)
    REAL TIME
    EXTERNAL XTIMSER
    SI1 = XTIMSER(3,66,0,TIME)
    RETURN
    END
C of SI1
C
    REAL FUNCTION SI2(TIME)
    REAL TIME
    EXTERNAL XTIMSER
    SI2 = XTIMSER(3,67,0,TIME)
    RETURN
    END
C of SI2
C
    REAL FUNCTION SI3(TIME)
    REAL TIME
    EXTERNAL XTIMSER
    SI3 = XTIMSER(3,68,0,TIME)
    RETURN
    END
C of SI3
C
    REAL FUNCTION SI4(TIME)
    REAL TIME
    EXTERNAL XTIMSER
    SI4 = XTIMSER(3,69,0,TIME)
    RETURN
    END
C of SI4
C
    REAL FUNCTION SI5(TIME)
    REAL TIME
    EXTERNAL XTIMSER
    SI5 = XTIMSER(3,70,0,TIME)
    RETURN
    END

```

```

C of SI5
C
    REAL FUNCTION SI6(TIME)
    REAL TIME
    EXTERNAL XTIMSER
    SI6 = XTIMSER(3,71,0,TIME)
    RETURN
    END
C of SI6
C
    REAL FUNCTION SI7(TIME)
    REAL TIME
    EXTERNAL XTIMSER
    SI7 = XTIMSER(3,72,0,TIME)
    RETURN
    END
C of SI7
C
    REAL FUNCTION SI8(TIME)
    REAL TIME
    EXTERNAL XTIMSER
    SI8 = XTIMSER(3,73,0,TIME)
    RETURN
    END
C of SI8

```

7.6 XTART.FOR

```
C#####
C SENECA 1.85 (C) NI00-CEM0/DGW
C File: XSTART.FOR
C Date: 1-7-92
C#####
      SUBROUTINE XSTART(TIME)
C
      IMPLICIT REAL(A-Z)
C Parameter:
      REAL TIME
C
C This routine will be called once at the begin of a simulation run,
C after all initializations but before the first results at TIME = 0
C are stored.
C-----
C Declarations:
      INCLUDE 'TEXAS.DCS'
      INCLUDE 'TEXAS.DCP'
      INCLUDE 'TEXAS.DCV'
      INCLUDE 'XSIM0.DEX'

      real dep(8),epi(8),j
c      integer j
      equivalence (dep(1),dep1)
      equivalence (epi(1),epi1)
C
C-----
C Statements:
C
      j=bay

c      do 10 j=1,8
      dep(j)=dep(j)*inidep(j)
      epi(j)=epi(j)*iniepi(j)

      if(dep(j).gt.1.e10.or.epi(j).gt.1e10)then
      dep(j)=xdiv(1,0)
      else
      endif

c10      continue

      END
C End of XSTART
C
C
C*****
      SUBROUTINE XEND(TIME)
C
      IMPLICIT REAL(A-Z)
C Parameter:
      REAL TIME
C
```

```

C This routine will be called once at the end of a simulation run,
C after the last calls to the submodel routines but before the last
C results are stored.
C-----
C Declarations:
    INCLUDE 'TEXAS.DCS'
    INCLUDE 'TEXAS.DCP'
    INCLUDE 'TEXAS.DCV'
    INCLUDE 'XSIM0.DEX'
C
C-----
C Statements:
C
    END
C End of XEND
C*****

```

7.7 XCAL.FOR

```
C#####
C SENECA 2.0 (C) NI00-CEM0/DGW
C File: XCAL.FOR
C Date: 19-1-93
C Version: 4
C This file contains the main program for the calibration program
C and the subroutines that are specific to the calibration program.
C#####
      PROGRAM XCAL
C-----
C Declarations:
C
      INTEGER ERRCODE,ERRIND,TLCODE,TYPRUN,TUNIT,TZERO,
      & FU,FURPR,FUSTR,FUVAR,FUPAR,FUBND,FUWST,FUFRC,FURAN,FUDAT,FURES,
      & FUTLG,FULOG,
      & FRSRES,FRSDAT,FPIRES,FPRRES,FNVRES,FRSRAN,FPTRES,FPTRAN,LOGMODE,
      & NSVAR,NBOUND,NWASTE,NFORC,NOUTVAR,NRANPAR,NVALVAR,MVALVAR,
      & NTLVAR,IOUTSTEP,IDRLABCH
      REAL DELT,DELTA,MODTIME,MXABSCHG,MNCHANGE,MNSWITCH,MNABSCHG,
      & MXRLABCH,NOVALUE,MAXVALUE
      COMMON /XCBSIM/ ERRCODE,ERRIND,TLCODE,TYPRUN,TUNIT,TZERO,
      & FU,FURPR,FUSTR,FUVAR,FUPAR,FUBND,FUWST,FUFRC,FURAN,FUDAT,FURES,
      & FUTLG,FULOG,
      & FRSRES,FRSDAT,FPIRES,FPRRES,FNVRES,FRSRAN,FPTRES,FPTRAN,LOGMODE,
      & NSVAR,NBOUND,NWASTE,NFORC,NOUTVAR,NRANPAR,NVALVAR,MVALVAR,
      & NTLVAR,IOUTSTEP,IDRLABCH,
      & DELT,DELTA,MODTIME,MXABSCHG,MNCHANGE,MNSWITCH,MNABSCHG,
      & MXRLABCH,NOVALUE,MAXVALUE
C
      INTEGER XMXGOF
      PARAMETER (XMXGOF = 8)
      INTEGER TSTART(0:6),RPI02,TSTOP(0:6),ITEND,
      & OUTFACT,NOUTSTEP,MXSTEPS,RPI05,GOFRES,GOFVAR,MNVALSTEP,NBAND,
      & IRUN,NRANRUN,NCALRUN,IBAND,NCENTR,ISEED,IBCONT,STOPCODE,
      & FIDENT,INTEGRATE,TIMEOFF,RPI09,IBACTPAR,RPI10(2:4)
      REAL
MXDELT,MNDELT,MXCHANGE,MXSWITCH,MNRELCHG,RPR02(2:4),RPR03(4),
      & GOFERR(XMXGOF)
      COMMON /XCBRPR/ TSTART,RPI02,TSTOP,ITEND,
      & OUTFACT,NOUTSTEP,MXSTEPS,RPI05,GOFRES,GOFVAR,MNVALSTEP,NBAND,
      & IRUN,NRANRUN,NCALRUN,IBAND,NCENTR,ISEED,IBCONT,STOPCODE,
      & FIDENT,INTEGRATE,TIMEOFF,RPI09,IBACTPAR,RPI10,
      & MXDELT,MNDELT,MXCHANGE,MXSWITCH,MNRELCHG,RPR02,RPR03,GOFERR
C
      INTEGER TYPSIM,TYPSENS,TYPCAL
      PARAMETER (TYPSIM = 1)
      PARAMETER (TYPSENS = 2)
      PARAMETER (TYPCAL = 3)
C-----
C Statements:
C
C Initialize calibration and read run parameters
      CALL XINSIM(TYPCAL)
```

```

        IF (ERRCODE .NE. 0) GOTO 100
C
C If proceeding calibration and initial are done
        IF ((IBCONT .NE. 0) .AND. (IRUN .GE. NRANRUN)) THEN
C then read random parameters from calibration result file
        CALL XOPRNDPAR(FURES)
        ELSE
C else read random parameters from random input file
        CALL XOPRNDPAR(FURAN)
        ENDIF
        IF (ERRCODE .NE. 0) GOTO 100
C
C Initialize variables, parameters and forcing functions
        CALL XRDSTRUCT
        IF (ERRCODE .NE. 0) GOTO 100
C
C Do calibration
        CALL XCALIBRATE
C
100  CALL XSTOP
        END
C End of program XCAL
C
C
C*****
        SUBROUTINE XOPRNDPAR(FUR)
C Parameter:
C
C File unit/type number
        INTEGER FUR
C
C Routine reads names of random parameters from
C random input or calibration result file.
C-----
C Declarations:
        INCLUDE 'TEXAS.DCM'
C
        INTEGER ERRCODE,ERRIND,TLCODE,TYPRUN,TUNIT,TZERO,
& FU,FURPR,FUSTR,FUVAR,FUPAR,FUBND,FUWST,FUFRC,FURAN,FUDAT,FURES,
& FUTLG,FULOG,
& FRSRES,FRSDAT,FPIRES,FPRRES,FNVRES,FRSRAN,FPTRES,FPTRAN,LOGMODE,
& NSVAR,NBOUND,NWASTE,NFORC,NOUTVAR,NRANPAR,NVALVAR,MVALVAR,
& NTLVAR,IOUTSTEP,IDRLABCH
        REAL DELT,DELTA,MODTIME,MXABSCG,MNCHANGE,MNSWITCH,MNABSCG,
& MXRLABCH,NOVALUE,MAXVALUE
        COMMON /XCBSIM/ ERRCODE,ERRIND,TLCODE,TYPRUN,TUNIT,TZERO,
& FU,FURPR,FUSTR,FUVAR,FUPAR,FUBND,FUWST,FUFRC,FURAN,FUDAT,FURES,
& FUTLG,FULOG,
& FRSRES,FRSDAT,FPIRES,FPRRES,FNVRES,FRSRAN,FPTRES,FPTRAN,LOGMODE,
& NSVAR,NBOUND,NWASTE,NFORC,NOUTVAR,NRANPAR,NVALVAR,MVALVAR,
& NTLVAR,IOUTSTEP,IDRLABCH,
& DELT,DELTA,MODTIME,MXABSCG,MNCHANGE,MNSWITCH,MNABSCG,
& MXRLABCH,NOVALUE,MAXVALUE
C
        INTEGER XMXGOF
        PARAMETER (XMXGOF = 8)
        INTEGER TSTART(0:6),RPI02,TSTOP(0:6),ITEND,

```

```

& OUTFACT, NOUTSTEP, MXSTEPS, RPI05, GOFRES, GOFVAR, MNVALSTEP, NBAND,
& IRUN, NRANRUN, NCALRUN, IBAND, NCENTR, ISEED, IBCONT, STOPCODE,
& FIDENT, INTEGRATE, TIMEOFF, RPI09, IBACTPAR, RPI10(2:4)
REAL
MXDELT, MNDELT, MXCHANGE, MXSWITCH, MNRELCHG, RPR02(2:4), RPR03(4),
& GOFERR(XMXGOF)
COMMON /XCBRPR/ TSTART, RPI02, TSTOP, ITEND,
& OUTFACT, NOUTSTEP, MXSTEPS, RPI05, GOFRES, GOFVAR, MNVALSTEP, NBAND,
& IRUN, NRANRUN, NCALRUN, IBAND, NCENTR, ISEED, IBCONT, STOPCODE,
& FIDENT, INTEGRATE, TIMEOFF, RPI09, IBACTPAR, RPI10,
& MXDELT, MNDELT, MXCHANGE, MXSWITCH, MNRELCHG, RPR02, RPR03, GOFERR
C
CHARACTER*16 NMMODEL, NVMODEL, NMTRES, NMERROR
CHARACTER*16 NMFRPR, NMFSTR, NMFVAR, NMFVAR, NMFVAR, NMFBNBND, NMFWSST, NMFFRC,
& NMFRES, NMFRAN, NMFDAT, NMFTLAG, NMFLOG
CHARACTER*78 OUTTXT
CHARACTER*160 ERRTXT
COMMON /XCBNMS/ NMMODEL, NVMODEL, NMTRES, NMERROR, NMFRPR, NMFSTR,
& NMFVAR, NMFVAR, NMFBNBND, NMFWSST, NMFFRC, NMFRES, NMFRAN, NMFDAT, NMFTLAG,
& NMFLOG, OUTTXT, ERRTXT
C
CHARACTER*16 NMOUTVAR(XMXOUTVAR), NMRANPAR(XMXRANPAR)
COMMON /XCBNAME/ NMOUTVAR, NMRANPAR
C
INTEGER IDOUTVAR(XMXOUTVAR), ISOUTVAR(XMXOUTVAR),
& IDRANPAR(XMXRANPAR), ISRANPAR(XMXRANPAR)
COMMON /XCBNMI/ IDOUTVAR, ISOUTVAR, IDRANPAR, ISRANPAR
C
REAL PARVEC(XMXRANPAR), LOCVEC(XMXRANPAR), PARDIS(XMXRANPAR),
& PARMIN(XMXRANPAR), PARMAX(XMXRANPAR)
COMMON /XCBRAN/ PARVEC, LOCVEC, PARDIS, PARMIN, PARMAX
C
INTEGER RCINT, RCREAL
PARAMETER (RCINT = 10, RCREAL = 2)
INTEGER FRSHHEAD, FRSTAIL, FRSTLAG
PARAMETER (FRSHHEAD = 16, FRSTAIL = 8, FRSTLAG = 4)
C
INTEGER I, IDUM, IVAR, RNDRUN, IDENT, NRTXT, NRINT, NRREAL
INTEGER POSINT, POSREAL, POSVARS
CHARACTER*16 NAME
C-----
C Statements:
C
C Initialization
DO 10 I=1, XMXRANPAR
IDRANPAR(I) = 0
10 CONTINUE
C
C Open random input file / calibration result file
CALL XOPFILE(FUR, FRSHHEAD, .TRUE., .TRUE., 1)
C and read first two records of header
CALL XRDHEAD(FRSRAN, NRTXT, NRINT, NRREAL, NRANPAR,
& POSINT, POSREAL, POSVARS, FPTRAN)
IF (ERRCODE .NE. 0) RETURN
C
C Check file consistency
IF (NRINT .LT. 1) THEN

```

```

        CALL XERROR(105)
        RETURN
    ENDIF
C
C Check number of random parameters in file
    IF (NRANPAR .LE. 0) THEN
        CALL XERROR(250)
        RETURN
    ELSEIF (NRANPAR .GT. XMXRANPAR) THEN
        ERRIND = XMXRANPAR - NRANPAR
        CALL XERROR(251)
        RETURN
    ENDIF
C
C Read number of random runs (RNDRUN) and file identification number
(IDENT)
    READ(FU,REC=POSINT+7,ERR=102,IOSTAT=ERRIND) IDUM,RNDRUN
    READ(FU,REC=POSINT+9,ERR=102,IOSTAT=ERRIND) IDENT
C
C If not proceeding calibration
    IF (IBCONT .EQ. 0) THEN
C Store number of random runs and file identification number
C in run parameters
        NRANRUN = RNDRUN
        FIDENT = IDENT
C If proceeding calibration
    ELSE
C Check file identification and number of random runs
        IF ((IDENT .NE. FIDENT) .OR. (RNDRUN .NE. NRANRUN) .OR.
&          (NRANPAR .NE. FNVRES)) THEN
            CALL XERROR(130)
            RETURN
        ENDIF
    ENDIF
C
C Read names of random parameters
    DO 20 IVAR = 1,NRANPAR
        READ(FU,REC=POSVARS+IVAR,ERR=102,IOSTAT=ERRIND) NAME
        NMRANPAR(IVAR) = NAME
20    CONTINUE
C
C Close file
    CLOSE(FU)
C and re-open with (new) tail record size
    CALL XOPFILE(FUR,FRSRAN,.TRUE.,.TRUE.,0)
C
C Read distribution codes of random parameters
    READ(FU,REC=FPTRAN+1,ERR=102,IOSTAT=ERRIND)
&      (PARDIS(I),I=1,NRANPAR)
C Read minimum values for random parameters
    READ(FU,REC=FPTRAN+2,ERR=102,IOSTAT=ERRIND)
&      (PARMIN(I),I=1,NRANPAR)
C Read maximum values for random parameters
    READ(FU,REC=FPTRAN+3,ERR=102,IOSTAT=ERRIND)
&      (PARMAX(I),I=1,NRANPAR)
C
C Close file

```

```

        CLOSE(FU)
C
C Check ranges random parameters
      DO 30 I=1,NRANPAR
        IF (PARMIN(I) .GE. PARMAX(I)) THEN
          NMERROR = NMRANPAR(I)
          CALL XERROR(253)
          RETURN
        ENDIF
30    CONTINUE
C
C Index sort random parameter names
      CALL XSORTNAMES(XMXRANPAR,NMRANPAR,NRANPAR,ISRANPAR)
C
      RETURN
C
102  CALL XERROR(102)
      END
C End of XOPRNDPAR
C
C
C*****
      SUBROUTINE XRDRNDPAR
C
C Routine reads irun-th random parameter vector from random input
C file, and stores random values in parameter common block.
C-----
C Declarations:
      INCLUDE 'TEXAS.DCS'
      INCLUDE 'TEXAS.DCM'
C
      INTEGER ERRCODE,ERRIND,TLCODE,TYPRUN,TUNIT,TZERO,
& FU,FURPR,FUSTR,FUVAR,FUPAR,FUBND,FUWST,FUFRC,FURAN,FUDAT,FURES,
& FUTLG,FULOG,
& FRSRES,FRSDAT,FPIRES,FPRRES,FNVRES,FRSRAN,FPTRES,FPTRAN,LOGMODE,
& NSVAR,NBOUND,NWASTE,NFORC,NOUTVAR,NRANPAR,NVALVAR,MVALVAR,
& NTLVAR,IOUTSTEP,IDRLABCH
      REAL DELT,DELTA,MODTIME,MXABSCHG,MNCHANGE,MNSWITCH,MNABSCHG,
& MXRLABCH,NOVALUE,MAXVALUE
      COMMON /XCBSIM/ ERRCODE,ERRIND,TLCODE,TYPRUN,TUNIT,TZERO,
& FU,FURPR,FUSTR,FUVAR,FUPAR,FUBND,FUWST,FUFRC,FURAN,FUDAT,FURES,
& FUTLG,FULOG,
& FRSRES,FRSDAT,FPIRES,FPRRES,FNVRES,FRSRAN,FPTRES,FPTRAN,LOGMODE,
& NSVAR,NBOUND,NWASTE,NFORC,NOUTVAR,NRANPAR,NVALVAR,MVALVAR,
& NTLVAR,IOUTSTEP,IDRLABCH,
& DELT,DELTA,MODTIME,MXABSCHG,MNCHANGE,MNSWITCH,MNABSCHG,
& MXRLABCH,NOVALUE,MAXVALUE
C
      INTEGER XMXGOF
      PARAMETER (XMXGOF = 8)
      INTEGER TSTART(0:6),RPI02,TSTOP(0:6),ITEND,
& OUTFACT,NOUTSTEP,MXSTEPS,RPI05,GOFRES,GOFVAR,MNVALSTEP,NBAND,
& IRUN,NRANRUN,NCALRUN,IBAND,NCENTR,ISEED,IBCONT,STOPCODE,
& FIDENT,INTEGRATE,TIMEOFF,RPI09,IBACTPAR,RPI10(2:4)
      REAL
MXDELT,MNDELT,MXCHANGE,MXSWITCH,MNRELCHG,RPR02(2:4),RPR03(4),
& GOFERR(XMXGOF)

```

```

COMMON /XCBRPR/ TSTART, RPI02, TSTOP, ITEND,
& OUTFACT, NOUTSTEP, MXSTEPS, RPI05, GOFRES, GOFVAR, MNVALSTEP, NBAND,
& IRUN, NRANRUN, NCALRUN, IBAND, NCENTR, ISEED, IBCONT, STOPCODE,
& FIDENT, INTEGRATE, TIMEOFF, RPI09, IBACTPAR, RPI10,
& MXDELT, MNDELT, MXCHANGE, MXSWITCH, MNRELCHG, RPR02, RPR03, GOFERR
C
CHARACTER*16 NMMODEL, NVMODEL, NMTRES, NMERROR
CHARACTER*16 NMFRPR, NMFSTR, NMFVAR, NMFPAR, NMFBND, NMFWST, NMFFRC,
& NMFRES, NMFRAN, NMFDAT, NMFTLAG, NMFLOG
CHARACTER*78 OUTTXT
CHARACTER*160 ERRTXT
COMMON /XCBNMS/ NMMODEL, NVMODEL, NMTRES, NMERROR, NMFRPR, NMFSTR,
& NMFVAR, NMFPAR, NMFBND, NMFWST, NMFFRC, NMFRES, NMFRAN, NMFDAT, NMFTLAG,
& NMFLOG, OUTTXT, ERRTXT
C
CHARACTER*16 NMOUTVAR(XMXOUTVAR), NMRANPAR(XMXRANPAR)
COMMON /XCBNAME/ NMOUTVAR, NMRANPAR
C
INTEGER IDOUTVAR(XMXOUTVAR), ISOUTVAR(XMXOUTVAR),
& IDRANPAR(XMXRANPAR), ISRANPAR(XMXRANPAR)
COMMON /XCBNMI/ IDOUTVAR, ISOUTVAR, IDRANPAR, ISRANPAR
C
REAL PARVEC(XMXRANPAR), LOCVEC(XMXRANPAR), PARDIS(XMXRANPAR),
& PARMIN(XMXRANPAR), PARMAX(XMXRANPAR)
COMMON /XCBRAN/ PARVEC, LOCVEC, PARDIS, PARMIN, PARMAX
C
REAL PAR(XMXPAR)
COMMON /XCBPAR/ PAR
C
INTEGER I
C-----
C Statements:
C
C Open random input file
CALL XOPFILE(FURAN, FRSRAN, .TRUE., .TRUE., 0)
IF (ERRCODE .NE. 0) RETURN
C
C Read irun-th random parameter vector
READ(FU, REC=FPTRAN+4+IRUN, ERR=102, IOSTAT=ERRIND)
& (PARVEC(I), I=1, NRANPAR)
C
C Close random input file
CLOSE(FU)
C
C For all random parameters
DO 10 I=1, NRANPAR
IF (IDRANPAR(I) .GT. 0) THEN
C Check value random parameter with range
IF ((PARVEC(I) .GE. PARMIN(I)) .AND.
& (PARVEC(I) .LE. PARMAX(I))) THEN
C Set actual value of parameter
PAR(IDRANPAR(I)) = PARVEC(I)
ELSE
NMERROR = NMRANPAR(I)
CALL XERROR(254)
RETURN
ENDIF

```

```

        ENDIF
10    CONTINUE
C
        RETURN
C
102    CALL XERROR(102)
        END
C End of XRDRNDPAR
C
C
C*****
        SUBROUTINE XOPCALPAR
C
C Routine creates calibration result file,
C and writes the run parameters, names of the random parameters,
C and the distribution codes, minimum and maximum values of the
C random parameters to the result file.
C-----
C Declarations:
        INCLUDE 'TEXAS.DCM'
C
        INTEGER ERRCODE,ERRIND,TLCODE,TYPRUN,TUNIT,TZERO,
        & FU,FURPR,FUSTR,FUVAR,FUPAR,FUBND,FUWST,FUFRC,FURAN,FUDAT,FURES,
        & FUTLG,FULOG,
        & FRSRES,FRSDAT,FPIRES,FPRRES,FNVRES,FRSRAN,FPTRES,FPTRAN,LOGMODE,
        & NSVAR,NBOUND,NWASTE,NFORC,NOUTVAR,NRANPAR,NVALVAR,MVALVAR,
        & NTLVAR,IOUTSTEP,IDRLABCH
        REAL DELT,DELTA,MODTIME,MXABSCHG,MNCHANGE,MNSWITCH,MNABSCHG,
        & MXRLABCH,NOVALUE,MAXVALUE
        COMMON /XCBSIM/ ERRCODE,ERRIND,TLCODE,TYPRUN,TUNIT,TZERO,
        & FU,FURPR,FUSTR,FUVAR,FUPAR,FUBND,FUWST,FUFRC,FURAN,FUDAT,FURES,
        & FUTLG,FULOG,
        & FRSRES,FRSDAT,FPIRES,FPRRES,FNVRES,FRSRAN,FPTRES,FPTRAN,LOGMODE,
        & NSVAR,NBOUND,NWASTE,NFORC,NOUTVAR,NRANPAR,NVALVAR,MVALVAR,
        & NTLVAR,IOUTSTEP,IDRLABCH,
        & DELT,DELTA,MODTIME,MXABSCHG,MNCHANGE,MNSWITCH,MNABSCHG,
        & MXRLABCH,NOVALUE,MAXVALUE
C
        INTEGER XMXGOF
        PARAMETER (XMXGOF = 8)
        INTEGER TSTART(0:6),RPI02,TSTOP(0:6),ITEND,
        & OUTFACT,NOUTSTEP,MXSTEPS,RPI05,GOFRES,GOFVAR,MNVALSTEP,NBAND,
        & IRUN,NRANRUN,NCALRUN,IBAND,NCENTR,ISEED,IBCONT,STOPCODE,
        & FIDENT,INTEGRATE,TIMEOFF,RPI09,IBACTPAR,RPI10(2:4)
        REAL
MXDELT,MNDELT,MXCHANGE,MXSWITCH,MNRELCHG,RPR02(2:4),RPR03(4),
        & GOFERR(XMXGOF)
        COMMON /XCBRPR/ TSTART,RPI02,TSTOP,ITEND,
        & OUTFACT,NOUTSTEP,MXSTEPS,RPI05,GOFRES,GOFVAR,MNVALSTEP,NBAND,
        & IRUN,NRANRUN,NCALRUN,IBAND,NCENTR,ISEED,IBCONT,STOPCODE,
        & FIDENT,INTEGRATE,TIMEOFF,RPI09,IBACTPAR,RPI10,
        & MXDELT,MNDELT,MXCHANGE,MXSWITCH,MNRELCHG,RPR02,RPR03,GOFERR
C
        CHARACTER*16 NMMODEL,NVMODEL,NMTRES,NMERROR
        CHARACTER*16 NMFRPR,NMFSTR,NMFVAR,NMFPAR,NMFBND,NMFWST,NMFFRC,
        & NMFRES,NMFRAN,NMFDAT,NMFTLAG,NMFLOG
        CHARACTER*78 OUTTXT

```

```

        CHARACTER*160 ERRTXT
        COMMON /XCBNMS/ NMMODEL, NVMODEL, NMTRES, NMERROR, NMFRPR, NMFSTR,
& NMFVAR, NMFPAR, NMFBND, NMFWST, NMFFRC, NMFRES, NMFRAN, NMFDAT, NMFTLAG,
& NMFLOG, OUTTXT, ERRTXT
C
        CHARACTER*16 NMOUTVAR(XMXOUTVAR), NMRANPAR(XMXRANPAR)
        COMMON /XCBNAME/ NMOUTVAR, NMRANPAR
C
        INTEGER IDOUTVAR(XMXOUTVAR), ISOUTVAR(XMXOUTVAR),
& IDRANPAR(XMXRANPAR), ISRANPAR(XMXRANPAR)
        COMMON /XCBNMI/ IDOUTVAR, ISOUTVAR, IDRANPAR, ISRANPAR
C
        REAL PARVEC(XMXRANPAR), LOCVEC(XMXRANPAR), PARDIS(XMXRANPAR),
& PARMIN(XMXRANPAR), PARMAX(XMXRANPAR)
        COMMON /XCBRAN/ PARVEC, LOCVEC, PARDIS, PARMIN, PARMAX
C
        INTEGER RCINT, RCREAL
        PARAMETER (RCINT = 10, RCREAL = 2)
        INTEGER FRSHEAD, FRSTAIL, FRSTLAG
        PARAMETER (FRSHEAD = 16, FRSTAIL = 8, FRSTLAG = 4)
C
        INTEGER I, NREC, POSVARS
C-----
C Statements:
C
C Calculate tail record size of calibration result file
        FRSRES = (NRANPAR+NBAND)*4
C
C Open calibration result file (with header record size),
C and write run parameters to result file
        CALL XWRRPR(FURES, FRSRES, NRANPAR, POSVARS, FPTRES, NMTRES)
C
C Write names of random parameters to result file
        DO 10 I=1, NRANPAR
            WRITE(FU, REC=POSVARS+I, ERR=102, IOSTAT=ERRIND) NMRANPAR(I)
10    CONTINUE
C
C Fill remainder of header until start of tail-part with dummy stars
        POSVARS = POSVARS + NRANPAR
        NREC = FRSRES/FRSHEAD + 1
        DO 20 I=1, NREC
            WRITE(FU, REC=POSVARS+I, ERR=102, IOSTAT=ERRIND) '*****'
20    CONTINUE
C
C Close result file
        CLOSE(FU)
C and re-open with (new) tail record size
        CALL XOPFILE(FURES, FRSRES, .FALSE., .TRUE., 0)
C
C Write distribution codes of random parameters to result file
        WRITE(FU, REC=FPTRES+1, ERR=102, IOSTAT=ERRIND)
& (PARDIS(I), I=1, NRANPAR)
C Write minimum values of random parameters to result file
        WRITE(FU, REC=FPTRES+2, ERR=102, IOSTAT=ERRIND)
& (PARMIN(I), I=1, NRANPAR)
C Write maximum values of random parameters to result file
        WRITE(FU, REC=FPTRES+3, ERR=102, IOSTAT=ERRIND)

```

```

      &      (PARMAX(I), I=1, NRANPAR)
C
C Close result file
      CLOSE(FU)
C
      RETURN
C
102  CALL XERROR(102)
      END
C End of XOPCALPAR
C
C
C*****
      SUBROUTINE XWRCALPAR(IDVEC)
C Parameter:
C
C Index number in vase/calibration result file
      INTEGER IDVEC
C
C Routine writes the values in random parameter vector PARVEC
C to the calibration result file in vase record IDVEC
C-----
C Declarations:
      INCLUDE 'TEXAS.DCM'
C
      INTEGER ERRCODE, ERRIND, TLCODE, TYPRUN, TUNIT, TZERO,
& FU, FURPR, FUSTR, FUVAR, FUPAR, FUBND, FUWST, FUFRC, FURAN, FUDAT, FURES,
& FUTLG, FULOG,
& FRSRES, FRSDAT, FPIRES, FPRRES, FNVRES, FRSRAN, FPTRES, FPTRAN, LOGMODE,
& NSVAR, NBOUND, NWASTE, NFORC, NOUTVAR, NRANPAR, NVALVAR, MVALVAR,
& NTLVAR, IOUTSTEP, IDRLABCH
      REAL DELT, DELTA, MODTIME, MXABSCG, MNCHANGE, MNSWITCH, MNABSCG,
& MXRLABCH, NOVALUE, MAXVALUE
      COMMON /XCBSIM/ ERRCODE, ERRIND, TLCODE, TYPRUN, TUNIT, TZERO,
& FU, FURPR, FUSTR, FUVAR, FUPAR, FUBND, FUWST, FUFRC, FURAN, FUDAT, FURES,
& FUTLG, FULOG,
& FRSRES, FRSDAT, FPIRES, FPRRES, FNVRES, FRSRAN, FPTRES, FPTRAN, LOGMODE,
& NSVAR, NBOUND, NWASTE, NFORC, NOUTVAR, NRANPAR, NVALVAR, MVALVAR,
& NTLVAR, IOUTSTEP, IDRLABCH,
& DELT, DELTA, MODTIME, MXABSCG, MNCHANGE, MNSWITCH, MNABSCG,
& MXRLABCH, NOVALUE, MAXVALUE
C
      INTEGER XMXGOF
      PARAMETER (XMXGOF = 8)
      INTEGER TSTART(0:6), RPI02, TSTOP(0:6), ITEND,
& OUTFACT, NOUTSTEP, MXSTEPS, RPI05, GOFRES, GOFVAR, MNVALSTEP, NBAND,
& IRUN, NRANRUN, NCALRUN, IBAND, NCENTR, ISEED, IBCONT, STOPCODE,
& FIDENT, INTEGRATE, TIMEOFF, RPI09, IBACTPAR, RPI10(2:4)
      REAL
MXDELT, MNDELT, MXCHANGE, MXSWITCH, MNRELCHG, RPR02(2:4), RPR03(4),
& GOFERR(XMXGOF)
      COMMON /XCBRPR/ TSTART, RPI02, TSTOP, ITEND,
& OUTFACT, NOUTSTEP, MXSTEPS, RPI05, GOFRES, GOFVAR, MNVALSTEP, NBAND,
& IRUN, NRANRUN, NCALRUN, IBAND, NCENTR, ISEED, IBCONT, STOPCODE,
& FIDENT, INTEGRATE, TIMEOFF, RPI09, IBACTPAR, RPI10,
& MXDELT, MNDELT, MXCHANGE, MXSWITCH, MNRELCHG, RPR02, RPR03, GOFERR
C

```

```

        INTEGER IDMNGOF, IDMXGOF
        REAL GOFNORM(XMXGOF+1), GOFWEIGHT(XMXOUTVAR),
& GOFVAL(XMXOUTVAR, XMXGOF),
& OBSDAT(XMXOUTSTEP), OBSERR(XMXOUTSTEP),
& OLDRES(0:XMXOUTVAR), CUMRES(0:XMXOUTVAR),
& RESULT(0:XMXOUTVAR, 0:XMXOUTSTEP+1)
        COMMON /XCBRES/ IDMNGOF, IDMXGOF, GOFNORM, GOFWEIGHT, GOFVAL,
& OBSDAT, OBSERR, OLDRES, CUMRES, RESULT
C
        REAL PARVEC(XMXRANPAR), LOCVEC(XMXRANPAR), PARDIS(XMXRANPAR),
& PARMIN(XMXRANPAR), PARMAX(XMXRANPAR)
        COMMON /XCBRAN/ PARVEC, LOCVEC, PARDIS, PARMIN, PARMAX
C
        INTEGER I
C-----
C Statements:
C
C Open calibration result file
        CALL XOPFILE(FURES, FRSRES, .TRUE., .TRUE., 0)
C
C Write parameter vector to result file in record IDVEC
        WRITE(FU, REC=FPTRES+3+IDVEC, ERR=102, IOSTAT=ERRIND)
& (PARVEC(I), I=1, NRANPAR), (GOFNORM(I), I=1, NBAND)
C
C Close result file
        CLOSE(FU)
C
C Write run number to result file
        CALL XWRRUN
C
        RETURN
C
102 CALL XERROR(102)
        END
C End of XWRCALPAR
C
C
C*****
        SUBROUTINE XRDCALPAR
C
C Routine calculates new random parameter vector out of the
C random parameter vectors in the vase, according to the
C controlled random search method as described in the manual.
C-----
C Declarations:
        INCLUDE 'TEXAS.DCS'
        INCLUDE 'TEXAS.DCM'
C
        INTEGER ERRCODE, ERRIND, TLCODE, TYPRUN, TUNIT, TZERO,
& FU, FURPR, FUSTR, FUVAR, FUPAR, FUBND, FUWST, FUFRC, FURAN, FUDAT, FURES,
& FUTLG, FULOG,
& FRSRES, FRSDAT, FPIRES, FPRRES, FNVRES, FRSRAN, FPTRES, FPTRAN, LOGMODE,
& NSVAR, NBOUND, NWASTE, NFORC, NOUTVAR, NRANPAR, NVALVAR, MVALVAR,
& NTLVAR, IOUTSTEP, IDRLABCH
        REAL DELT, DELTA, MODTIME, MXABSCG, MNCHANGE, MNSWITCH, MNABSCG,
& MXRLABCH, NOVALUE, MAXVALUE
        COMMON /XCBSIM/ ERRCODE, ERRIND, TLCODE, TYPRUN, TUNIT, TZERO,

```

```

& FU, FURPR, FUSTR, FUVAR, FUPAR, FUBND, FUWST, FUFRC, FURAN, FUDAT, FURES,
& FUTLG, FULOG,
& FRSRES, FRSDAT, FPIRES, FPRRES, FNVRES, FRSRAN, FPTRES, FPTRAN, LOGMODE,
& NSVAR, NBOUND, NWASTE, NFORC, NOUTVAR, NRANPAR, NVALVAR, MVALVAR,
& NTLVAR, IOUTSTEP, IDRLABCH,
& DELT, DELTA, MODTIME, MXABSCG, MNCHANGE, MNSWITCH, MNABSCG,
& MXRLABCH, NOVALUE, MAXVALUE
C
  INTEGER XMXGOF
  PARAMETER (XMXGOF = 8)
  INTEGER TSTART(0:6), RPI02, TSTOP(0:6), ITEND,
&  OUTFACT, NOUTSTEP, MXSTEPS, RPI05, GOFRES, GOFVAR, MNVALSTEP, NBAND,
&  IRUN, NRANRUN, NCALRUN, IBAND, NCENTR, ISEED, IBCONT, STOPCODE,
&  FIDENT, INTEGRATE, TIMEOFF, RPI09, IBACTPAR, RPI10(2:4)
  REAL
MXDELT, MNDELT, MXCHANGE, MXSWITCH, MNRELCHG, RPR02(2:4), RPR03(4),
&  GOFERR(XMXGOF)
  COMMON /XCBPR/ TSTART, RPI02, TSTOP, ITEND,
&  OUTFACT, NOUTSTEP, MXSTEPS, RPI05, GOFRES, GOFVAR, MNVALSTEP, NBAND,
&  IRUN, NRANRUN, NCALRUN, IBAND, NCENTR, ISEED, IBCONT, STOPCODE,
&  FIDENT, INTEGRATE, TIMEOFF, RPI09, IBACTPAR, RPI10,
&  MXDELT, MNDELT, MXCHANGE, MXSWITCH, MNRELCHG, RPR02, RPR03, GOFERR
C
  REAL PAR(XMXPAR)
  COMMON /XCBPAR/ PAR
C
  CHARACTER*16 NMOUTVAR(XMXOUTVAR), NMRANPAR(XMXRANPAR)
  COMMON /XCBNAME/ NMOUTVAR, NMRANPAR
C
  INTEGER IDOUTVAR(XMXOUTVAR), ISOUTVAR(XMXOUTVAR),
&  IDRANPAR(XMXRANPAR), ISRANPAR(XMXRANPAR)
  COMMON /XCBNMI/ IDOUTVAR, ISOUTVAR, IDRANPAR, ISRANPAR
C
  REAL PARVEC(XMXRANPAR), LOCVEC(XMXRANPAR), PARDIS(XMXRANPAR),
&  PARMIN(XMXRANPAR), PARMAX(XMXRANPAR)
  COMMON /XCBRAN/ PARVEC, LOCVEC, PARDIS, PARMIN, PARMAX
C
  INTEGER IDMNCAL, IDMXCAL, MNCENTR, MXCENTR
  REAL CALNORM(XMXVASE)
  COMMON /XCBCAL/ IDMNCAL, IDMXCAL, MNCENTR, MXCENTR, CALNORM
C
  INTEGER I, IREC, IVASE, VASE(XMXVASE)
  REAL RANGE, SUMVEC(XMXRANPAR)
C-----
C Statements:
C
C Open calibration result file
  CALL XOPFILE(FURES,FRSRES,.TRUE.,.TRUE.,0)
C
C Initialise sum array
  DO 10 I=1,NRANPAR
    SUMVEC(I) = 0.0
10  CONTINUE
C
C Get random permutation over NRANRUN
  CALL XGETVASE(NRANRUN,VASE)
C

```

```

C Read at random NCENTR parameter vectors from vase
C and calculate centroid
    IREC = FPTRES + 3
    DO 30 IVASE=1,NCENTR
        READ(FU,REC=IREC+VASE(IVASE),ERR=102,IOSTAT=ERRIND)
        &      (LOCVEC(I),I=1,NRANPAR)
        DO 20 I=1,NRANPAR
            SUMVEC(I) = SUMVEC(I) + LOCVEC(I)
20      CONTINUE
30      CONTINUE
C
C Final calculation centroid
    DO 40 I=1,NRANPAR
        SUMVEC(I) = SUMVEC(I)/NCENTR
40      CONTINUE
C
C Read additional parameter vector from vase
    READ(FU,REC=IREC+VASE(NCENTR+1),ERR=102,IOSTAT=ERRIND)
    &      (LOCVEC(I),I=1,NRANPAR)
C
C Close result file
    CLOSE(FU)
C
C Mirror parameter vector at centroid
    DO 50 I=1,NRANPAR
        PARVEC(I) = SUMVEC(I) + SUMVEC(I) - LOCVEC(I)
C Check vector space
    RANGE = PARMAX(I) - PARMIN(I)
C If parameter value out of range then
C map value in range
    IF (PARVEC(I) .LT. PARMIN(I)+0.005*RANGE) THEN
        PARVEC(I) = PARMIN(I) - (0.005/1.005) *
        &      (PARMIN(I) - RANGE - PARVEC(I))
    ELSEIF (PARVEC(I) .GT. PARMAX(I)-0.005*RANGE) THEN
        PARVEC(I) = PARMAX(I) - (0.005/1.005) *
        &      (PARMAX(I) + RANGE - PARVEC(I))
    ENDIF
C Store new value for parameter in parameter common block
    PAR(IDRANPAR(I)) = PARVEC(I)
50      CONTINUE
C
    RETURN
C
102      CALL XERROR(102)
    END
C End of XRDICALPAR
C
C
C*****
C      SUBROUTINE XRDNORM
C
C Routine determines range of random parameter values and
C Goodness of Fit values (for active error band) in vase,
C and reads parameter vector with lowest Goodness of Fit
C value from vase.
C-----
C Declarations:

```

```

      INCLUDE 'TEXAS.DCM'

C
      INTEGER ERRCODE, ERRIND, TLCODE, TYPRUN, TUNIT, TZERO,
& FU, FURPR, FUSTR, FUVAR, FUPAR, FUBND, FUWST, FUFRC, FURAN, FUDAT, FURES,
& FUTLG, FULOG,
& FRSRES, FRSDAT, FPIRES, FPRRES, FNVRES, FRSRAN, FPTRES, FPTRAN, LOGMODE,
& NSVAR, NBOUND, NWASTE, NFORC, NOUTVAR, NRANPAR, NVALVAR, MVALVAR,
& NTLVAR, IOUTSTEP, IDRLABCH
      REAL DELT, DELTA, MODTIME, MXABSCG, MNCHANGE, MNSWITCH, MNABSCG,
& MXRLABCH, NOVALUE, MAXVALUE
      COMMON /XCBSIM/ ERRCODE, ERRIND, TLCODE, TYPRUN, TUNIT, TZERO,
& FU, FURPR, FUSTR, FUVAR, FUPAR, FUBND, FUWST, FUFRC, FURAN, FUDAT, FURES,
& FUTLG, FULOG,
& FRSRES, FRSDAT, FPIRES, FPRRES, FNVRES, FRSRAN, FPTRES, FPTRAN, LOGMODE,
& NSVAR, NBOUND, NWASTE, NFORC, NOUTVAR, NRANPAR, NVALVAR, MVALVAR,
& NTLVAR, IOUTSTEP, IDRLABCH,
& DELT, DELTA, MODTIME, MXABSCG, MNCHANGE, MNSWITCH, MNABSCG,
& MXRLABCH, NOVALUE, MAXVALUE

C
      INTEGER XMXGOF
      PARAMETER (XMXGOF = 8)
      INTEGER TSTART(0:6), RPI02, TSTOP(0:6), ITEND,
& OUTFACT, NOUTSTEP, MXSTEPS, RPI05, GOFRES, GOFVAR, MNVALSTEP, NBAND,
& IRUN, NRANRUN, NCALRUN, IBAND, NCENTR, ISEED, IBCONT, STOPCODE,
& FIDENT, INTEGRATE, TIMEOFF, RPI09, IBACTPAR, RPI10(2:4)
      REAL
MXDELT, MNDELT, MXCHANGE, MXSWITCH, MNRELCHG, RPR02(2:4), RPR03(4),
& GOFERR(XMXGOF)
      COMMON /XCBRPR/ TSTART, RPI02, TSTOP, ITEND,
& OUTFACT, NOUTSTEP, MXSTEPS, RPI05, GOFRES, GOFVAR, MNVALSTEP, NBAND,
& IRUN, NRANRUN, NCALRUN, IBAND, NCENTR, ISEED, IBCONT, STOPCODE,
& FIDENT, INTEGRATE, TIMEOFF, RPI09, IBACTPAR, RPI10,
& MXDELT, MNDELT, MXCHANGE, MXSWITCH, MNRELCHG, RPR02, RPR03, GOFERR

C
      REAL PARVEC(XMXRANPAR), LOCVEC(XMXRANPAR), PARDIS(XMXRANPAR),
& PARMIN(XMXRANPAR), PARMAX(XMXRANPAR)
      COMMON /XCBRAN/ PARVEC, LOCVEC, PARDIS, PARMIN, PARMAX

C
      INTEGER IDMNAL, IDMXCAL, MNCENTR, MXCENTR
      REAL CALNORM(XMXVASE)
      COMMON /XCBCAL/ IDMNAL, IDMXCAL, MNCENTR, MXCENTR, CALNORM

C
      INTEGER I, IREC, IVASE
      REAL NORM(XMXGOF)

C-----
C Statements:
C
C Open calibration result file
      CALL XOPFILE(FURES, FRSRES, .TRUE., .TRUE., 0)
C
C Read all parameter vectors from vase
      IREC = FPTRES + 3
      DO 10 IVASE=1, NRANRUN
C Read ivase-th parameter vector
      READ(FU, REC=IREC+IVASE, ERR=102, IOSTAT=ERRIND)
& (LOCVEC(I), I=1, NRANPAR), (NORM(I), I=1, NBAND)
C Store Goodness of Fit value of active error band

```

```

        CALNORM(IVASE) = NORM(IBAND)
10    CONTINUE
C
C Calculate minimum and maximum of Goodness of Fit values in vase
      CALL XMINMAX(NRANRUN,CALNORM,NOVALUE,IDMNCAL,IDMXCAL)
C
C Read parameter vector with lowest Goodness of Fit value from vase
      READ(FU,REC=IREC+IDMNCAL,ERR=102,IOSTAT=ERRIND)
      &      (LOCVEC(I),I=1,NRANPAR)
C
C Close result file
      CLOSE(FU)
C
      RETURN
C
102  CALL XERROR(102)
      END
C End of XRDNORM
C
C
C*****
      SUBROUTINE XGETVASE(NVASE,VASE)
C Parameters:
C
C Number of values in VASE
      INTEGER NVASE
C Returns array with permutation
      INTEGER VASE(NVASE)
C
C Routine generates permutation over NVASE and returns
C permutation in VASE.
C-----
C Declarations:
      INCLUDE 'TEXAS.DCM'
C
      INTEGER XMXGOF
      PARAMETER (XMXGOF = 8)
      INTEGER TSTART(0:6),RPI02,TSTOP(0:6),ITEND,
      & OUTFACT,NOUTSTEP,MXSTEPS,RPI05,GOFRES,GOFVAR,MNVALSTEP,NBAND,
      & IRUN,NRANRUN,NCALRUN,IBAND,NCENTR,ISEED,IBCONT,STOPCODE,
      & FIDENT,INTEGRATE,TIMEOFF,RPI09,IBACTPAR,RPI10(2:4)
      REAL
MXDELT,MNDELT,MXCHANGE,MXSWITCH,MNRELCHG,RPR02(2:4),RPR03(4),
      & GOFERR(XMXGOF)
      COMMON /XCBRPR/ TSTART,RPI02,TSTOP,ITEND,
      & OUTFACT,NOUTSTEP,MXSTEPS,RPI05,GOFRES,GOFVAR,MNVALSTEP,NBAND,
      & IRUN,NRANRUN,NCALRUN,IBAND,NCENTR,ISEED,IBCONT,STOPCODE,
      & FIDENT,INTEGRATE,TIMEOFF,RPI09,IBACTPAR,RPI10,
      & MXDELT,MNDELT,MXCHANGE,MXSWITCH,MNRELCHG,RPR02,RPR03,GOFERR
C
      EXTERNAL XRAN
      INTEGER I
      REAL RVASE(XMXVASE)
C-----
C Statements:
C
C Generate array with random values

```

```

        DO 10 I=1,NVASE
            RVASE(I) = XTRAN(ISEED)
10      CONTINUE
C
C Index sort random array
C so that the index array will be a permutation over NVASE
        CALL XSORTREALS(NVASE,RVASE,VASE)
C
        END
C End of XGETVASE
C
C
C*****
        REAL FUNCTION XTRAN(ISEED)
C Parameter:
C
C Seed for generator
        INTEGER ISEED
C
C Simple random value generator
C-----
C Declarations:
        INTEGER IM,IA,IC
        REAL RM
        PARAMETER (IM=259200,IA=7141,IC=54773, RM=1.0/IM)
C-----
C Statements:
C
        ISEED = MOD(ISEED*IA+IC,IM)
        XTRAN = FLOAT(ISEED)*RM
C
        END
C End of XTRAN
C
C
C*****
        SUBROUTINE XCALIBRATE
C
C Main driver routine for calibration.
C-----
C Declarations:
        INCLUDE 'TEXAS.DCS'
        INCLUDE 'TEXAS.DCM'
C
        INTEGER ERRCODE,ERRIND,TLCODE,TYPRUN,TUNIT,TZERO,
& FU,FURPR,FUSTR,FUVAR,FUPAR,FUBND,FUWST,FUFRC,FURAN,FUDAT,FURES,
& FUTLG,FULOG,
& FRSRES,FRSDAT,FPIRES,FPRRES,FNVRES,FRSRAN,FPTRES,FPTRAN,LOGMODE,
& NSVAR,NBOUND,NWASTE,NFORC,NOUTVAR,NRANPAR,NVALVAR,MVALVAR,
& NTLVAR,IOUTSTEP,IDRLABCH
        REAL DELT,DELTA,MODTIME,MXABSCHG,MNCHANGE,MNSWITCH,MNABSCHG,
& MXRLABCH,NOVALUE,MAXVALUE
        COMMON /XCBSIM/ ERRCODE,ERRIND,TLCODE,TYPRUN,TUNIT,TZERO,
& FU,FURPR,FUSTR,FUVAR,FUPAR,FUBND,FUWST,FUFRC,FURAN,FUDAT,FURES,
& FUTLG,FULOG,
& FRSRES,FRSDAT,FPIRES,FPRRES,FNVRES,FRSRAN,FPTRES,FPTRAN,LOGMODE,
& NSVAR,NBOUND,NWASTE,NFORC,NOUTVAR,NRANPAR,NVALVAR,MVALVAR,

```

```

& NTLVAR, IOUTSTEP, IDRLABCH,
& DELT, DELTA, MODTIME, MXABSCHG, MNCHANGE, MNSWITCH, MNABSCHG,
& MXRLABCH, NOVALUE, MAXVALUE
C
  INTEGER XMXGOF
  PARAMETER (XMXGOF = 8)
  INTEGER TSTART(0:6), RPI02, TSTOP(0:6), ITEND,
&  OUTFACT, NOUTSTEP, MXSTEPS, RPI05, GOFRES, GOFVAR, MNVALSTEP, NBAND,
&  IRUN, NRANRUN, NCALRUN, IBAND, NCENTR, ISEED, IBCONT, STOPCODE,
&  FIDENT, INTEGRATE, TIMEOFF, RPI09, IBACTPAR, RPI10(2:4)
  REAL
MXDELT, MNDELT, MXCHANGE, MXSWITCH, MNRELCHG, RPR02(2:4), RPR03(4),
&  GOFERR(XMXGOF)
  COMMON /XCBRPR/ TSTART, RPI02, TSTOP, ITEND,
&  OUTFACT, NOUTSTEP, MXSTEPS, RPI05, GOFRES, GOFVAR, MNVALSTEP, NBAND,
&  IRUN, NRANRUN, NCALRUN, IBAND, NCENTR, ISEED, IBCONT, STOPCODE,
&  FIDENT, INTEGRATE, TIMEOFF, RPI09, IBACTPAR, RPI10,
&  MXDELT, MNDELT, MXCHANGE, MXSWITCH, MNRELCHG, RPR02, RPR03, GOFERR
C
  CHARACTER*16 NMMODEL, NVMODEL, NMTRES, NMERROR
  CHARACTER*16 NMFRPR, NMFSTR, NMFVAR, NMFPAR, NMFBND, NMFWST, NMFFRC,
&  NMFRES, NMFRAN, NMFDAT, NMFTLAG, NMFLOG
  CHARACTER*78 OUTTXT
  CHARACTER*160 ERRTXT
  COMMON /XCBNMS/ NMMODEL, NVMODEL, NMTRES, NMERROR, NMFRPR, NMFSTR,
&  NMFVAR, NMFPAR, NMFBND, NMFWST, NMFFRC, NMFRES, NMFRAN, NMFDAT, NMFTLAG,
&  NMFLOG, OUTTXT, ERRTXT
C
  CHARACTER*16 NMOUTVAR(XMXOUTVAR), NMRANPAR(XMXRANPAR)
  COMMON /XCBNAME/ NMOUTVAR, NMRANPAR
C
  INTEGER IDOUTVAR(XMXOUTVAR), ISOUTVAR(XMXOUTVAR),
&  IDRANPAR(XMXRANPAR), ISRANPAR(XMXRANPAR)
  COMMON /XCBNMI/ IDOUTVAR, ISOUTVAR, IDRANPAR, ISRANPAR
C
  INTEGER IDMNGOF, IDMXGOF
  REAL GOFNORM(XMXGOF+1), GOFWEIGHT(XMXOUTVAR),
&  GOFVAL(XMXOUTVAR, XMXGOF),
&  OBSDAT(XMXOUTSTEP), OBSERR(XMXOUTSTEP),
&  OLDRES(0:XMXOUTVAR), CUMRES(0:XMXOUTVAR),
&  RESULT(0:XMXOUTVAR, 0:XMXOUTSTEP+1)
  COMMON /XCBRES/ IDMNGOF, IDMXGOF, GOFNORM, GOFWEIGHT, GOFVAL,
&  OBSDAT, OBSERR, OLDRES, CUMRES, RESULT
C
  REAL PARVEC(XMXRANPAR), LOCVEC(XMXRANPAR), PARDIS(XMXRANPAR),
&  PARMIN(XMXRANPAR), PARMAX(XMXRANPAR)
  COMMON /XCBRAN/ PARVEC, LOCVEC, PARDIS, PARMIN, PARMAX
C
  INTEGER IDMNAL, IDMXCAL, MNCENTR, MXCENTR
  REAL CALNORM(XMXVASE)
  COMMON /XCBAL/ IDMNAL, IDMXCAL, MNCENTR, MXCENTR, CALNORM
C
  REAL PAR(XMXPAR)
  COMMON /XCBPAR/ PAR
C
  REAL DVAR(XMXSVAR), VAR(XMXVAR)
  COMMON /XCBVAR/ DVAR, VAR

```

```

C
    INTEGER I
    LOGICAL REPLACED, FIRSTRUN
    REAL SAVGOF, SAVVAR(XMXVAR)
C-----
C Statements:
C
C Check number of vectors in vase
    IF (NRANRUN .LT. 20) THEN
        ERRIND = 20
        CALL XERROR(260)
        RETURN
    ENDIF
    IF (NRANRUN .GT. XMXVASE) THEN
        ERRIND = XMXVASE - NRANRUN
        CALL XERROR(261)
        RETURN
    ENDIF
C
C Check number of output variables with observed data
C for calculating Goodness of Fit
    IF (NVALVAR .LE. 0) THEN
        CALL XERROR(406)
        RETURN
    ENDIF
C
C Determine working range for Centroid size
    MNCENTR = NRANRUN/5 + 1
    MXCENTR = (4*NRANRUN)/5
C
C If not proceeding calibration then
    IF (IBCONT .EQ. 0) THEN
C initialize Centroid size
        NCENTR = MNCENTR
C Create calibration result file
        CALL XOPCALPAR
C If proceeding calibration then
    ELSE
C Check stop code
        IF (STOPCODE .GT. 0) THEN
            CALL XWARNING(121)
            RETURN
        ENDIF
        CALL XWARNING(120)
    ENDIF
C
C Initializations
    STOPCODE = 0
C Save initial values variables
    DO 10 I=1, XMXVAR
        SAVVAR(I) = VAR(I)
10    CONTINUE
    FIRSTRUN = .TRUE.
C
C If initial runs have already been done then skip initial runs
    IF (IBAND .GT. 0) GOTO 99
C

```

```

C Initial runs loop:
C
C Check for fatal error
C Don't stop on non-fatal error
20  IF ((ERRCODE .GE. 100) .OR. (ERRCODE .LT. 0)) RETURN
C
C Reset error code
      ERRCODE = 0
C
C If last initial run has been done the goto next phase
      IF (IRUN .GE. NRANRUN) THEN
          IBAND = NBAND
          IRUN = 0
          GOTO 99
      ENDIF
C
C Display number of run message on screen
      WRITE(*,997) IRUN+1,NRANRUN
997  FORMAT(44X,'INITIAL RUN =',I6,' of ',I6)
C
C Restore initial values of the variables:
      DO 30 I=1,XXVAR
          VAR(I) = SAVVAR(I)
30  CONTINUE
C
C Read random values for the parameters from random input file
      CALL XRDRNDPAR
C
C Do a simulation run
      CALL XSIMULATE
C
C Check if values of random parameters have been changed
      IF ((ERRCODE .EQ. 0) .AND. FIRSTRUN) THEN
          FIRSTRUN = .FALSE.
          DO 40 I=1,NRANPAR
              IF (PAR(IDRANPAR(I)) .NE. PARVEC(I)) THEN
                  NMERROR = NMRANPAR(I)
                  CALL XERROR(257)
              ENDIF
40  CONTINUE
      ENDIF
C
C Write parameter vector to vase/result file
      CALL XWRCALPAR(IRUN)
C
C Goto begin of initial runs loop
      GOTO 20
C End of initial runs loop
99  CONTINUE
C
C Determine range of Goodness of Fit values in vase
      CALL XRDNORM
C
C Controlled random search runs loop
C
C Check for fatal error
C Don't stop on non-fatal error

```

```

100  IF ((ERRCODE .GE. 100) .OR. (ERRCODE .LT. 0)) RETURN
C
C Reset error code
      ERRCODE = 0
C
C Check if ready with active error band
C Active error band is ready when the worst Goodness of Fit is zero
105  IF ((IDMXCAL .GT. 0) .AND. (CALNORM(IDMXCAL) .LE. 0.0)) THEN
C
C If last (most inner) error band is done the calibration is ready
      IF (IBAND .LE. 1) THEN
          WRITE(*,999) ' !! READY with GoF band: ',GOFERR(1)
          STOPCODE = 3
          RETURN
      ENDIF
C
C Decrease number of active error band
      IBAND = IBAND - 1
C Reset run number counter (counts runs per error band)
      IRUN = 0
C Reset Centroid size
      NCENTR = MNCENTR
C
C Display start of error band message on screen
      WRITE(*,999) ' !! Starting GoF band: ',GOFERR(IBAND)
C
C Determine range of Goodness of Fit values in vase for new error band
      CALL XRDNORM
C
C Goto error band check for new error band
      GOTO 105
      ENDIF
C
C If number of runs for error band exceeds allowed number of runs
C then stop calibration
      IF (IRUN .GE. NCALRUN) THEN
          STOPCODE = -3
          RETURN
      ENDIF
C
C Display calibration run message on screen
      WRITE(*,998) IBAND,IRUN+1,NCALRUN
998  FORMAT(44X,'BAND',I2,' RUN =',I6,' of ',I6)
C
C Restore initial values of the variables:
      DO 110 I=1,XXVAR
          VAR(I) = SAVVAR(I)
110  CONTINUE
C
C Generate new random values for random parameters
      CALL XRDICALPAR
C
C Do a simulation run
      CALL XSIMULATE
C
C Check if values of random parameters have been changed
      IF ((ERRCODE .EQ. 0) .AND. FIRSTRUN) THEN

```

```

        FIRSTRUN = .FALSE.
        DO 140 I=1,NRANPAR
            IF (PAR(IDRANPAR(I)) .NE. PARVEC(I)) THEN
                NMERROR = NMRANPAR(I)
                CALL XERROR(257)
            ENDIF
140      CONTINUE
        ENDIF
C
C Check if Goodness of Fit of run is better (lower) than worst
C Goodness of Fit in vase
        REPLACED = .FALSE.
        IF (ERRCODE .EQ. 0) THEN
C Check Goodness of Fit value
            IF ((GOFNORM(IBAND) .LT. CALNORM(IDMXCAL)) .AND.
                & (GOFNORM(IBAND+1) .EQ. 0.0)) THEN
C If Goodness of Fit for run = 0 then increase centroid size
C for less randomness in Centroid -> faster convergence
                IF (GOFNORM(IBAND) .EQ. 0.0) THEN
                    IF (NCENTR .LT. MXCENTR) NCENTR = NCENTR + 1
                ENDIF
C
C Replace worst Goodness of Fit in vase with new Goodness of Fit value
                SAVGOF = CALNORM(IDMXCAL)
                CALNORM(IDMXCAL) = GOFNORM(IBAND)
C
C Write parameter vector of better run to vase/result file
                CALL XWRCALPAR(IDMXCAL)
C
C Determine new best/worst Goodness of Fit values in vase
                CALL XMINMAX(NRANRUN, CALNORM, NOVALUE, IDMNICAL, IDMXCAL)
C
C Display success messages on screen
                WRITE(*,999)
                & ' SUCCESSFUL !! Replacing GoF(',GOFERR(IBAND),'): ',SAVGOF
                REPLACED = .TRUE.
            ENDIF
        ENDIF
C
C If not better run then write only number of run to result file
        IF (.NOT. REPLACED) CALL XWRRUN
C
C Continue with calibration loop
        GOTO 100
C End of calibration loop.
C
999  FORMAT (A,F5.2,A,G13.5)
        END
C End of XCALIBRATE
C
C#### End of file #####

```

7.8 XSENS.FOR

```
C#####
C SENECA 2.0 (C) NI00-CEM0/DGW
C File: XSENS.FOR
C Date: 17-1-93
C Version: 4
C This file contains the main program for the sensitivity analysis
program
C and the subroutines that are specific to the sensitivity analysis
program.
C#####
      PROGRAM XSENS
C-----
C Declarations:
C
      INTEGER ERRCODE,ERRIND,TLCODE,TYPRUN,TUNIT,TZERO,
      & FU,FURPR,FUSTR,FUVAR,FUPAR,FUBND,FUWST,FUFRC,FURAN,FUDAT,FURES,
      & FUTLG,FULOG,
      & FRSRES,FRSDAT,FPIRES,FPRRES,FNVRES,FRSRAN,FPTRES,FPTRAN,LOGMODE,
      & NSVAR,NBOUND,NWASTE,NFORC,NOUTVAR,NRANPAR,NVALVAR,MVALVAR,
      & NTLVAR,IOUTSTEP,IDRLABCH
      REAL DELT,DELTA,MODTIME,MXABSCHG,MNCHANGE,MNSWITCH,MNABSCHG,
      & MXRLABCH,NOVALUE,MAXVALUE
      COMMON /XCBSIM/ ERRCODE,ERRIND,TLCODE,TYPRUN,TUNIT,TZERO,
      & FU,FURPR,FUSTR,FUVAR,FUPAR,FUBND,FUWST,FUFRC,FURAN,FUDAT,FURES,
      & FUTLG,FULOG,
      & FRSRES,FRSDAT,FPIRES,FPRRES,FNVRES,FRSRAN,FPTRES,FPTRAN,LOGMODE,
      & NSVAR,NBOUND,NWASTE,NFORC,NOUTVAR,NRANPAR,NVALVAR,MVALVAR,
      & NTLVAR,IOUTSTEP,IDRLABCH,
      & DELT,DELTA,MODTIME,MXABSCHG,MNCHANGE,MNSWITCH,MNABSCHG,
      & MXRLABCH,NOVALUE,MAXVALUE
C
      INTEGER TYPSIM,TYPSENS,TYPCAL
      PARAMETER (TYPSIM = 1)
      PARAMETER (TYPSENS = 2)
      PARAMETER (TYPCAL = 3)
C-----
C Statements:
C
C Initialize sensitivity analysis and read run parameters
      CALL XINSIM(TYPSENS)
      IF (ERRCODE .NE. 0) GOTO 100
C
C Read random parameters from random input file
      CALL XOPRANPAR(FURAN)
      IF (ERRCODE .NE. 0) GOTO 100
C
C Initialize variables, parameters and forcing functions
      CALL XRDSTRUCT
      IF (ERRCODE .NE. 0) GOTO 100
C
C Do sensitivity analysis runs
      CALL XMONTECARLO
C
```

```

100    CALL XSTOP
      END
C End of program XSSENS
C
C
C*****
      SUBROUTINE XOPRANPAR(FUR)
C Parameter:
C
C File unit/type number
      INTEGER FUR
C
C Routine reads names of random parameters from random input file.
C-----
C Declarations:
      INCLUDE 'TEXAS.DCM'
C
      INTEGER ERRCODE, ERRIND, TLCODE, TYPRUN, TUNIT, TZERO,
& FU, FURPR, FUSTR, FUVAR, FUPAR, FUBND, FUWST, FUFRC, FURAN, FUDAT, FURES,
& FUTLG, FULOG,
& FRSRES, FRSDAT, FPIRES, FPRRES, FNVRES, FRSRAN, FPTRES, FPTRAN, LOGMODE,
& NSVAR, NBOUND, NWASTE, NFORC, NOUTVAR, NRANPAR, NVALVAR, MVALVAR,
& NTLVAR, IOUTSTEP, IDRLABCH
      REAL DELT, DELTA, MODTIME, MXABSCG, MNCHANGE, MNSWITCH, MNABSCG,
& MXRLABCH, NOVALUE, MAXVALUE
      COMMON /XCBSIM/ ERRCODE, ERRIND, TLCODE, TYPRUN, TUNIT, TZERO,
& FU, FURPR, FUSTR, FUVAR, FUPAR, FUBND, FUWST, FUFRC, FURAN, FUDAT, FURES,
& FUTLG, FULOG,
& FRSRES, FRSDAT, FPIRES, FPRRES, FNVRES, FRSRAN, FPTRES, FPTRAN, LOGMODE,
& NSVAR, NBOUND, NWASTE, NFORC, NOUTVAR, NRANPAR, NVALVAR, MVALVAR,
& NTLVAR, IOUTSTEP, IDRLABCH,
& DELT, DELTA, MODTIME, MXABSCG, MNCHANGE, MNSWITCH, MNABSCG,
& MXRLABCH, NOVALUE, MAXVALUE
C
      INTEGER XMXGOF
      PARAMETER (XMXGOF = 8)
      INTEGER TSTART(0:6), RPI02, TSTOP(0:6), ITEND,
& OUTFACT, NOUTSTEP, MXSTEPS, RPI05, GOFRES, GOFVAR, MNVALSTEP, NBAND,
& IRUN, NRANRUN, NCALRUN, IBAND, NCENTR, ISEED, IBCONT, STOPCODE,
& FIDENT, INTEGRATE, TIMEOFF, RPI09, IBACTPAR, RPI10(2:4)
      REAL
MXDELT, MNDELT, MXCHANGE, MXSWITCH, MNRELCHG, RPR02(2:4), RPR03(4),
& GOFERR(XMXGOF)
      COMMON /XCBRPR/ TSTART, RPI02, TSTOP, ITEND,
& OUTFACT, NOUTSTEP, MXSTEPS, RPI05, GOFRES, GOFVAR, MNVALSTEP, NBAND,
& IRUN, NRANRUN, NCALRUN, IBAND, NCENTR, ISEED, IBCONT, STOPCODE,
& FIDENT, INTEGRATE, TIMEOFF, RPI09, IBACTPAR, RPI10,
& MXDELT, MNDELT, MXCHANGE, MXSWITCH, MNRELCHG, RPR02, RPR03, GOFERR
C
      CHARACTER*16 NMMODEL, NVMODEL, NMTRES, NMERROR
      CHARACTER*16 NMFRPR, NMFSTR, NMFVAR, NMFPAR, NMFBND, NMFWST, NMFFRC,
& NMFRES, NMFRAN, NMFDAT, NMFTLAG, NMFLOG
      CHARACTER*78 OUTTXT
      CHARACTER*160 ERRTXT
      COMMON /XCBNMS/ NMMODEL, NVMODEL, NMTRES, NMERROR, NMFRPR, NMFSTR,
& NMFVAR, NMFPAR, NMFBND, NMFWST, NMFFRC, NMFRES, NMFRAN, NMFDAT, NMFTLAG,
& NMFLOG, OUTTXT, ERRTXT

```

```

C      CHARACTER*16 NMOUTVAR(XMXOUTVAR),NMRANPAR(XMXRANPAR)
      COMMON /XCBNAME/ NMOUTVAR,NMRANPAR
C
      INTEGER IDOUTVAR(XMXOUTVAR),ISOUTVAR(XMXOUTVAR),
& IDRANPAR(XMXRANPAR),ISRANPAR(XMXRANPAR)
      COMMON /XCBNMI/ IDOUTVAR,ISOUTVAR,IDRANPAR,ISRANPAR
C
      REAL PARVEC(XMXRANPAR),LOCVEC(XMXRANPAR),PARDIS(XMXRANPAR),
& PARMIN(XMXRANPAR),PARMAX(XMXRANPAR)
      COMMON /XCBRAN/ PARVEC,LOCVEC,PARDIS,PARMIN,PARMAX
C
      INTEGER RCINT,RCREAL
      PARAMETER (RCINT = 10, RCREAL = 2)
      INTEGER FRSHED,FRSTAIL,FRSTLAG
      PARAMETER (FRSHED = 16, FRSTAIL = 8, FRSTLAG = 4)
C
      INTEGER I,IDUM,IVAR,RANRUN,IDENT
      INTEGER NRTXT,NRINT,NRREAL,POSINT,POSREAL,POSVARS
      CHARACTER*16 NAME
C-----
C Statements:
C
C Initialization
      DO 10 I=1,XMXRANPAR
          IDRANPAR(I) = 0
10    CONTINUE
C
C Open random input file
      CALL XOPFILE(FUR,FRSHED,.TRUE.,.TRUE.,1)
C and read first two records of header
      CALL XRDHEAD(FRSRAN,NRTXT,NRINT,NRREAL,NRANPAR,
& POSINT,POSREAL,POSVARS,FPTRAN)
      IF (ERRCODE .NE. 0) RETURN
C
C Check file consistency
      IF (NRINT .LT. 1) THEN
          CALL XERROR(105)
          RETURN
      ENDIF
C
C Check number of random parameters in file
      IF (NRANPAR .LE. 0) THEN
          CALL XERROR(250)
          RETURN
      ELSEIF (NRANPAR .GT. XMXRANPAR) THEN
          ERRIND = XMXRANPAR - NRANPAR
          CALL XERROR(251)
          RETURN
      ENDIF
C
C Read number of random runs (RANRUN) and file identification number
(IDENT)
      READ(FU,REC=POSINT+7,ERR=102,IOSTAT=ERRIND) IDUM,RANRUN
      READ(FU,REC=POSINT+9,ERR=102,IOSTAT=ERRIND) IDENT
C
C If not proceeding sensitivity

```

```

        IF (IBCONT .EQ. 0) THEN
C Store number of random runs and file identification number
C in run parameters
        NRANRUN = RANRUN
        FIDENT = IDENT
C If proceeding sensitivity
        ELSE
C Check file identification and number of random runs
        IF ((IDENT .NE. FIDENT) .OR. (RANRUN .NE. NRANRUN)) THEN
            CALL XERROR(130)
            RETURN
        ENDIF
    ENDIF
C
C Read names of random parameters
    DO 20 IVAR = 1, NRANPAR
        READ(FU, REC=POSVARS+IVAR, ERR=102, IOSTAT=ERRIND) NAME
        NMRANPAR(IVAR) = NAME
20    CONTINUE
C
C Close file
    CLOSE(FU)
C and re-open with (new) tail record size
    CALL XOPFILE(FUR, FRSRAN, .TRUE., .TRUE., 0)
C
C Read distribution codes of random parameters
    READ(FU, REC=FPTRAN+1, ERR=102, IOSTAT=ERRIND)
    &    (PARDIS(I), I=1, NRANPAR)
C Read minimum values for random parameters
    READ(FU, REC=FPTRAN+2, ERR=102, IOSTAT=ERRIND)
    &    (PARMIN(I), I=1, NRANPAR)
C Read maximum values for random parameters
    READ(FU, REC=FPTRAN+3, ERR=102, IOSTAT=ERRIND)
    &    (PARMAX(I), I=1, NRANPAR)
C
C Close file
    CLOSE(FU)
C
C Check ranges random parameters
    DO 30 I=1, NRANPAR
        IF (PARMIN(I) .GE. PARMAX(I)) THEN
            NMERROR = NMRANPAR(I)
            CALL XERROR(253)
            RETURN
        ENDIF
30    CONTINUE
C
C Index sort random parameter names
    CALL XSORTNAMES(XMXRANPAR, NMRANPAR, NRANPAR, ISRANPAR)
C
    RETURN
C
102    CALL XERROR(102)
    END
C End of XOPRANPAR
C
C

```

```

C*****
      SUBROUTINE XRDRANPAR
C
C Routine reads irun-th random parameter vector from random input
C file, and stores random values in parameter common block.
C-----
C Declarations:
      INCLUDE 'TEXAS.DCS'
      INCLUDE 'TEXAS.DCM'
C
      INTEGER ERRCODE,ERRIND,TLCODE,TYPRUN,TUNIT,TZERO,
& FU,FURPR,FUSTR,FUVAR,FUPAR,FUBND,FUWST,FUFRC,FURAN,FUDAT,FURES,
& FUTLG,FULOG,
& FRSRES,FRSDAT,FPIRES,FPRRES,FNVRES,FRSRAN,FPTRES,FPTRAN,LOGMODE,
& NSVAR,NBOUND,NWASTE,NFORC,NOUTVAR,NRANPAR,NVALVAR,MVALVAR,
& NTLVAR,IOUTSTEP,IDRLABCH
      REAL DELT,DELTA,MODTIME,MXABSCG,MNCHANGE,MNSWITCH,MNABSCG,
& MXRLABCH,NOVALUE,MAXVALUE
      COMMON /XCBSIM/ ERRCODE,ERRIND,TLCODE,TYPRUN,TUNIT,TZERO,
& FU,FURPR,FUSTR,FUVAR,FUPAR,FUBND,FUWST,FUFRC,FURAN,FUDAT,FURES,
& FUTLG,FULOG,
& FRSRES,FRSDAT,FPIRES,FPRRES,FNVRES,FRSRAN,FPTRES,FPTRAN,LOGMODE,
& NSVAR,NBOUND,NWASTE,NFORC,NOUTVAR,NRANPAR,NVALVAR,MVALVAR,
& NTLVAR,IOUTSTEP,IDRLABCH,
& DELT,DELTA,MODTIME,MXABSCG,MNCHANGE,MNSWITCH,MNABSCG,
& MXRLABCH,NOVALUE,MAXVALUE
C
      INTEGER XMXGOF
      PARAMETER (XMXGOF = 8)
      INTEGER TSTART(0:6),RPI02,TSTOP(0:6),ITEND,
& OUTFACT,NOUTSTEP,MXSTEPS,RPI05,GOFRES,GOFVAR,MNVALSTEP,NBAND,
& IRUN,NRANRUN,NCALRUN,IBAND,NCENTR,ISEED,IBCONT,STOPCODE,
& FIDENT,INTEGRATE,TIMEOFF,RPI09,IBACTPAR,RPI10(2:4)
      REAL
MXDELT,MNDELT,MXCHANGE,MXSWITCH,MNRELCHG,RPR02(2:4),RPR03(4),
& GOFERR(XMXGOF)
      COMMON /XCBRPR/ TSTART,RPI02,TSTOP,ITEND,
& OUTFACT,NOUTSTEP,MXSTEPS,RPI05,GOFRES,GOFVAR,MNVALSTEP,NBAND,
& IRUN,NRANRUN,NCALRUN,IBAND,NCENTR,ISEED,IBCONT,STOPCODE,
& FIDENT,INTEGRATE,TIMEOFF,RPI09,IBACTPAR,RPI10,
& MXDELT,MNDELT,MXCHANGE,MXSWITCH,MNRELCHG,RPR02,RPR03,GOFERR
C
      CHARACTER*16 NMMODEL,NVMODEL,NMTRES,NMERROR
      CHARACTER*16 NMFRPR,NMFSTR,NMFVAR,NMFPAR,NMFBND,NMFWST,NMFFRC,
& NMFRES,NMFRAN,NMFDAT,NMFTLAG,NMFLOG
      CHARACTER*78 OUTTXT
      CHARACTER*160 ERRTXT
      COMMON /XCBNMS/ NMMODEL,NVMODEL,NMTRES,NMERROR,NMFRPR,NMFSTR,
& NMFVAR,NMFPAR,NMFBND,NMFWST,NMFFRC,NMFRES,NMFRAN,NMFDAT,NMFTLAG,
& NMFLOG,OUTTXT,ERRTXT
C
      CHARACTER*16 NMOUTVAR(XMXOUTVAR),NMRANPAR(XMXRANPAR)
      COMMON /XCBNAME/ NMOUTVAR,NMRANPAR
C
      INTEGER IDOUTVAR(XMXOUTVAR),ISOUTVAR(XMXOUTVAR),
& IDRANPAR(XMXRANPAR),ISRANPAR(XMXRANPAR)
      COMMON /XCBNMI/ IDOUTVAR,ISOUTVAR,IDRANPAR,ISRANPAR

```

```

C      REAL PARVEC(XMXRANPAR), LOCVEC(XMXRANPAR), PARDIS(XMXRANPAR),
&      PARMIN(XMXRANPAR), PARMAX(XMXRANPAR)
      COMMON /XCBRAN/ PARVEC, LOCVEC, PARDIS, PARMIN, PARMAX
C
      REAL PAR(XMXPAR)
      COMMON /XCBPAR/ PAR
C
      INTEGER I
C-----
C Statements:
C
C Open random input file
      CALL XOPFILE(FURAN,FRSRAN,.TRUE.,.TRUE.,0)
      IF (ERRCODE .NE. 0) RETURN
C
C Read irun-th random parameter vector
      READ(FU,REC=FPTRAN+4+IRUN,ERR=102,IOSTAT=ERRIND)
&      (PARVEC(I),I=1,NRANPAR)
C
C Close random input file
      CLOSE(FU)
C
C For all random parameters
      DO 10 I = 1,NRANPAR
        IF (IDRANPAR(I) .GT. 0) THEN
C Check value random parameter with range
          IF ((PARVEC(I) .GE. PARMIN(I)) .AND.
&          (PARVEC(I) .LE. PARMAX(I))) THEN
C Set actual value of parameter
            PAR(IDRANPAR(I)) = PARVEC(I)
          ELSE
            NMERROR = NMRANPAR(I)
            CALL XERROR(254)
            RETURN
          ENDIF
        ENDIF
      CONTINUE
      RETURN
C
102  CALL XERROR(102)
      END
C End of XRDRANPAR
C
C
C*****
      SUBROUTINE XMONTECARLO
C
C Main driver routine for sensitivity analysis program.
C Uses Monte Carlo runs to asses influence of uncertain parameters
C on outcome of simulation model.
C-----
C Declarations:
      INCLUDE 'TEXAS.DCS'
      INCLUDE 'TEXAS.DCM'
C
      INTEGER ERRCODE,ERRIND,TLCODE,TYPRUN,TUNIT,TZERO,

```

```

& FU, FURPR, FUSTR, FUVAR, FUPAR, FUBND, FUWST, FUFRC, FURAN, FUDAT, FURES,
& FUTLG, FULOG,
& FRSRES, FRSDAT, FPIRES, FPRRES, FNVRES, FRSRAN, FPTRES, FPTRAN, LOGMODE,
& NSVAR, NBOUND, NWASTE, NFORC, NOUTVAR, NRANPAR, NVALVAR, MVALVAR,
& NTLVAR, IOUTSTEP, IDRLABCH
  REAL DELT, DELTA, MODTIME, MXABSCG, MNCHANGE, MNSWITCH, MNABSCG,
& MXRLABCH, NOVALUE, MAXVALUE
  COMMON /XCBSIM/ ERRCODE, ERRIND, TLCODE, TYPRUN, TUNIT, TZERO,
& FU, FURPR, FUSTR, FUVAR, FUPAR, FUBND, FUWST, FUFRC, FURAN, FUDAT, FURES,
& FUTLG, FULOG,
& FRSRES, FRSDAT, FPIRES, FPRRES, FNVRES, FRSRAN, FPTRES, FPTRAN, LOGMODE,
& NSVAR, NBOUND, NWASTE, NFORC, NOUTVAR, NRANPAR, NVALVAR, MVALVAR,
& NTLVAR, IOUTSTEP, IDRLABCH,
& DELT, DELTA, MODTIME, MXABSCG, MNCHANGE, MNSWITCH, MNABSCG,
& MXRLABCH, NOVALUE, MAXVALUE
C
  INTEGER XMXGOF
  PARAMETER (XMXGOF = 8)
  INTEGER TSTART(0:6), RPI02, TSTOP(0:6), ITEND,
& OUTFACT, NOUTSTEP, MXSTEPS, RPI05, GOFRES, GOFVAR, MNVALSTEP, NBAND,
& IRUN, NRANRUN, NCALRUN, IBAND, NCENTR, ISEED, IBCONT, STOPCODE,
& FIDENT, INTEGRATE, TIMEOFF, RPI09, IBACTPAR, RPI10(2:4)
  REAL
MXDELT, MNDELT, MXCHANGE, MXSWITCH, MNRELCHG, RPR02(2:4), RPR03(4),
& GOFERR(XMXGOF)
  COMMON /XCBRPR/ TSTART, RPI02, TSTOP, ITEND,
& OUTFACT, NOUTSTEP, MXSTEPS, RPI05, GOFRES, GOFVAR, MNVALSTEP, NBAND,
& IRUN, NRANRUN, NCALRUN, IBAND, NCENTR, ISEED, IBCONT, STOPCODE,
& FIDENT, INTEGRATE, TIMEOFF, RPI09, IBACTPAR, RPI10,
& MXDELT, MNDELT, MXCHANGE, MXSWITCH, MNRELCHG, RPR02, RPR03, GOFERR
C
  CHARACTER*16 NMMODEL, NVMODEL, NMTRES, NMERROR
  CHARACTER*16 NMFRPR, NMFSTR, NMFVAR, NMFPAR, NMFBND, NMFWST, NMFFRC,
& NMFRES, NMFRAN, NMFDAT, NMFTLAG, NMFLOG
  CHARACTER*78 OUTTXT
  CHARACTER*160 ERRTXT
  COMMON /XCBNMS/ NMMODEL, NVMODEL, NMTRES, NMERROR, NMFRPR, NMFSTR,
& NMFVAR, NMFPAR, NMFBND, NMFWST, NMFFRC, NMFRES, NMFRAN, NMFDAT, NMFTLAG,
& NMFLOG, OUTTXT, ERRTXT
C
  CHARACTER*16 NMOUTVAR(XMXOUTVAR), NMRANPAR(XMXRANPAR)
  COMMON /XCBNAME/ NMOUTVAR, NMRANPAR
C
  INTEGER IDOUTVAR(XMXOUTVAR), ISOUTVAR(XMXOUTVAR),
& IDRANPAR(XMXRANPAR), ISRANPAR(XMXRANPAR)
  COMMON /XCBNMI/ IDOUTVAR, ISOUTVAR, IDRANPAR, ISRANPAR
C
  REAL PARVEC(XMXRANPAR), LOCVEC(XMXRANPAR), PARDIS(XMXRANPAR),
& PARMIN(XMXRANPAR), PARMAX(XMXRANPAR)
  COMMON /XCBRAN/ PARVEC, LOCVEC, PARDIS, PARMIN, PARMAX
C
  REAL PAR(XMXPAR)
  COMMON /XCBPAR/ PAR
C
  REAL DVAR(XMXSVAR), VAR(XMXVAR)
  COMMON /XCBVAR/ DVAR, VAR
C

```

```

        INTEGER I
        LOGICAL FIRSTRUN
        REAL SAVVAR(XMXVAR)
C-----
C Statements:
C
C If not proceeding sensitivity analysis then create result file
        IF (IBCONT .EQ. 0) THEN
            CALL XOPRESULT
C If proceeding sensitivity analysis then check stop code
        ELSE
            IF ((STOPCODE .GT. 0) .OR. (IRUN .GE. NRANRUN)) THEN
                CALL XWARNING(121)
                STOPCODE = 2
                RETURN
            ENDIF
            CALL XWARNING(120)
        ENDIF
C
        STOPCODE = 0
C Save initial values variables
        DO 10 I=1,XMXVAR
            SAVVAR(I) = VAR(I)
10    CONTINUE
        FIRSTRUN = .TRUE.
C
C MONTE CARLO run loop:
C
C Check for fatal error
C Don't stop on non-fatal error
20    IF ((ERRCODE .GE. 100) .OR. (ERRCODE .LT. 0)) RETURN
C
C Reset error code
        ERRCODE = 0
C
C If all random runs done then ready with sensitivity analysis and stop
        IF (IRUN .GE. NRANRUN) THEN
            STOPCODE = 2
            RETURN
        ENDIF
C
C Display run number message on screen
        WRITE(*,999) IRUN+1,NRANRUN
999    FORMAT(44X,'SENSITIVITY RUN =',I5,' of ',I5)
C
C Restore initial values of the variables
        DO 30 I=1,XMXVAR
            VAR(I) = SAVVAR(I)
30    CONTINUE
C
C Read random values for the parameters from random input file
        CALL XRDRANPAR
C
C Do a simulation run
        CALL XSIMULATE
C
C Check if values of random parameters have been changed

```

```

        IF ((ERRCODE .EQ. 0) .AND. FIRSTRUN) THEN
            DO 40 I=1,NRANPAR
                IF (PAR(IDRANPAR(I)) .NE. PARVEC(I)) THEN
                    NMERROR = NMRANPAR(I)
                    CALL XERROR(257)
                ENDIF
40      CONTINUE
            ENDIF
C
C Write simulation run results to result file
        CALL XWRRESULT
C
C Continue with monte carlo loop
        GOTO 20
C
        END
C End of XMONTECARLO
C
C#### End of file #####

```

8. TWDB Review Comments



TEXAS WATER DEVELOPMENT BOARD



James E. Herring, *Chairman*
Lewis H. McMahan, *Member*
Edward G. Vaughan, *Member*

J. Kevin Ward
Executive Administrator

Jack Hunt, *Vice Chairman*
Thomas Weir Labatt III, *Member*
Joe M. Crutcher, *Member*

April 7, 2010

Paul Montagna, Ph.D.
Endowed Chair for Ecosystems Studies and Modeling
Harte Research Institute for Gulf of Mexico Studies
Texas A&M University - Corpus Christi
6300 Ocean Drive, Unit #5869
Corpus Christi, Texas 78412-5869

Re: Research Contract between the Texas Water Development Board (TWDB) and
Texas A&M University - Corpus Christi (TAMUCC), TWDB Contract No.
0804830791, Draft Final Report Comments

Dear Dr. Montagna:

Staff members of the TWDB have completed a review of the draft report prepared under the above-referenced contract. ATTACHMENT I provides the comments resulting from this review. As stated in the TWDB contract, TAMUCC will consider incorporating draft report comments from the EXECUTIVE ADMINISTRATOR as well as other reviewers into the final report. In addition, TAMUCC will include a copy of the EXECUTIVE ADMINISTRATOR'S draft report comments in the Final Report.

The TWDB looks forward to receiving one (1) electronic copy of the entire Final Report in Portable Document Format (PDF) and six (6) bound double-sided copies. TAMUCC shall also submit one (1) electronic copy of any computer programs or models, and, if applicable, an operations manual developed under the terms of this Contract.

If you have any questions concerning the contract, please contact Dr. Carla Guthrie, the TWDB's designated Contract Manager for this project at (512) 463-4179.

Sincerely,

Robert E. Mace, Ph.D., P.G.
Deputy Executive Administrator
Water Science and Conservation

Enclosures

c: Carla Guthrie, Ph.D., TWDB

Our Mission

To provide leadership, planning, financial assistance, information, and education for the conservation and responsible development of water for Texas.

P.O. Box 13231 • 1700 N. Congress Avenue • Austin, Texas 78711-3231
Telephone (512) 463-7847 • Fax (512) 475-2053 • 1-800-RELAYTX (for the hearing impaired)
www.twdb.state.tx.us • info@twdb.state.tx.us
TNRIS - Texas Natural Resources Information System • www.tnris.state.tx.us
A Member of the Texas Geographic Information Council (TGIC)



Effect of Climatic Variability on Freshwater Inflow, Benthic Communities, and Secondary Production in Texas Lagoonal Estuaries

**PI Paul Montagna, PhD
Contract number #0804830791**

TWDB comments to final draft

This study reflects the tasks that were outlined in the Scope of Work (SOW), with two exceptions: (1) The SOW identifies a 10-yr dataset for calibration and a 5-yr dataset for validation, but the study actually used an 11-yr dataset for calibration and a 20-yr dataset for validation. This change benefits the study by allowing for a longer period of record for model development and analysis. (2) The draft final report was submitted in March 2010 rather than the original target of December 2008 as identified in the SOW. A no-cost time extension was allowed due to unforeseeable delays.

The report was a good effort to refine an ecological model to predict system-wide secondary production. The deviations were at times large, but are expected from an ecological mechanistic model.

REQUIRED CHANGES

General Draft Final Report Comments:

The report states that model calibration was based on an 11-year dataset (1988 to 1999) and validation was based on a 20-year dataset (1988-2008). Since the 11-year dataset was used to generate parameter estimates for model simulation, the results of model simulations, using those parameters, should be compared only to data from 2000-2008. Simulation data for the period 1988-1999 is informative when compared to real data, but does not provide a true test of model performance. The report should either clarify the model calibration and validation effort or address this in the presentation of results and in the discussion of model performance.

Additionally, while the model shows promise, it has large amounts of unexplained variation. Please explain how this variation affects the estimates of benthic responses to changes in salinity shown in the Discussion section.

Overall, this is a well-structured report; however, please be sure to proofread the report, and correct spelling and grammatical errors.

Specific Draft Final Report Comments:

1. Introduction, Page 3, 1st paragraph, last sentence: Please add the (Montagna and Kalke 1995) citation to the Reference section at the end of the document.
2. Introduction, Page 3, 2nd paragraph, 3rd sentence: Please add the (Montagna and Li 1996) citation to the Reference section.

3. Methods, Section 2.1.1, Page 4: Location names in the text do not match those shown in Figure 1.
4. Methods, Section 2.1.3., Page 6, last paragraph: Table 2 only shows the correlations between salinity and nutrients, and does not show the correlation between salinity and inflow. Please add the salinity/inflow correlation statistics to the table, or revise the text.
5. Methods, Page 20, Section 2.4, second to last sentence: Please write out the acronym 'GoF.'
6. Methods, Page 21, 1st sentence: Please add (de Hoop et al. 1989) citation to the Reference section.
7. Methods, Section 2.5, Page 22, 2nd sentence: The Model Validation sections needs additional explanation about the model validation procedures. Additionally, please identify the April 1988-October 2008 data set that was used for validation.
8. Results, Section 3.1, Page 23, Table 3 and associated text: Is salinity and temperature statistically different between the primary and secondary bays? There may be a tendency for secondary bays to be lower, but the values don't look to be statistically different. Additionally, this is more true for salinity than for temperature. Temperature looks to be the same for all bays.
9. Results, Section 3.2, Page 24, Table 4: Some of the Best Fit values for Temperature seem unrealistically high. Note: one value is highlighted in red font. Please explain or correct if red color is unintended.
10. Discussion, Section 4.1, Page 40, 3rd sentence: Please add (O'Neill and Gardner, 1979 and Walters 1986) citations to the Reference section.
11. Discussion, Section 4.3, Page 45, Figure 15: It would be helpful to list the long-term mean salinity, which was used as the starting point for these graphics, for each bay. With a different salinity starting point, one should be able to compare how existing conditions in Bay A (e.g. 25psu) compare to a 20% change in Bay B which creates 25psu conditions. Do these two bays look similar when conditions are similar? Such questions could then be explored.
12. Discussion, Page 46, 2nd paragraph, last sentence: Be sure to complete the sentence or remove it.
13. Discussion, Page 47, Section 4.4, last sentence: Please add the (Li et al. 1996 and Montagna and Li 1997) citations to the Reference section.
14. Discussion, Page 48, 2nd paragraph, last sentence: Please add Rozas et al. (2005) to the Reference section.
15. Page 50: Please add the (Powell et al. 2002) citation to the Reference section.

SUGGESTED CHANGES

16. Page 8: Section 2.1.5: Consider describing the source of temperature and nutrient data that was used in creating a model for primary production.
17. Page 23, section 3.2: Consider mentioning a range of Percent RMS values that are considered acceptable for ecological models. It may be helpful to the reader to mention it here, or alert the reader that the topic will be discussed in more detail in the Discussion section.
18. Page 42, Section 4.3: Consider putting this information in the Results section since new results are being presented.