MODELING AND MONITORING LONG-TERM CHANGE IN MACROBENTHOS IN TEXAS ESTUARIES

by

Paul A. Montagna and Jian Li

UTMSI Technical Report TR/96-001

FINAL REPORT

MODELING AND MONITORING LONG-TERM CHANGE IN MACROBENTHOS IN TEXAS ESTUARIES

Paul A. Montagna, Ph.D.
and
Jian Li, Ph.D.

The University of Texas at Austin, Marine Science Institute
750 Channelview Drive, Port Aransas, Texas 78373
Telephone: (512)749-6779
Fax: (512)749-6777
E-mail: paul@utmsi.zo.utexas.edu

Submitted to:

Texas Water Development Board
Contract No. 96-483-132

UTMSI Technical Report Number 96 TR/96-001

September 1996

PROJECT SUMMARY

There is a need for the development of quantitative relationships between freshwater inflow and biotic indicator variables. While much of this kind of work has been done in the past, it was based on a regression approach and not a biological mechanistic approach. A bioenergetic model was developed that relates macrobenthic productivity to salinity within and among four Texas estuaries. The four Texas estuaries: the Lavaca-Colorado, Guadalupe, Nueces and Laguna Madre, lie in a climatic gradient with decreasing rainfall and concordant decreasing freshwater inflow. A long-term data set of macrobenthic biomass, which is an indicator of productivity, was used to calibrate the model. The benthos were divided into two trophic groups: deposit feeders (that consume detritus or sediment organic matter) and suspension feeders (that filter phytoplankton or graze on benthic diatoms). Variation within estuaries was so high that the model did a poor job of simulating long-term change of estuarine-wide benthic biomass. Each estuary is composed of a primary bay, which is nearest to the Gulf, and a secondary bay, which is nearest to the inflow source. Simulations for the eight Texas bays did fit the data well, indicating that the structure of Texas estuaries is strongly influenced by inflow and Gulf exchange. Within estuaries the production to biomass ratio (P/B), with units of 1/year, increased with proximity to the freshwater inflow source. The P/B ratio for deposit feeders generally increased with water residence time, i.e., inflow volume adjusted by the estuary volume, but declined with water residence time for suspension feeders. This trend is consistent with the hypothesis that suspension feeders are good indicators of the importance of freshwater inflow on maintaining secondary production. The one exception was Laguna Madre, with the highest P/B of 3.2, but this high value is due to extensive seagrass habitat, which is lacking in the other bays. Corpus Christi Bay had the lowest P/B of 1.2, and this is probably due to several forms of natural and anthropogenic disturbance.

ACKNOWLEDGMENTS

Funding for the modeling study was provided by the Texas Water Development Board's (TWDB) Water Research and Planning Fund, authorized under Texas Water code Sections 15.402 and 16.058(e), and administered by the Department under interagency cooperative contracts No. 96-483-132. The authors have benefitted by discussions and exchange of data with Gary Powell, William Longley, and David Brock of the TWDB.

Benthic data used to calibrate the simulations in this manuscript was obtained as a result of many different research projects, but the primary source of funding was through the TWDB. The research in the Lavaca-Colorado, Guadalupe and Nueces Estuaries was partially funded through the TWDB Water Research and Planning Fund, under interagency cooperative contracts Nos. 8-483-607, 9-483-705, 9-483-706, 93-483-352, 94-483-003, and 95-483-068.

Other partial support for work in Matagorda Bay was supplied by the Lower Colorado River Authority (LCRA). Work in the Nueces Estuary was partially funded by Institutional Grant NA16RG0445-01 to Texas A&M University Sea Grant Program from the National Sea Grant Office, National Oceanic and Atmospheric Administration, U.S. Department of Commerce. Partial support was provided for work in Laguna Madre and Baffin Bay by the Texas Higher Education Coordinating Board, Advanced Technology Program under Grant Nos. 4541 and 3658-264. Partial support was also provided by University of Texas at Austin, Marine Science Institute.

Mr. Lawrence McEachron, Texas Parks and Wildlife Department, was very helpful for obtaining the Coastal Fisheries monitoring data for the four estuaries.

The authors especially thank Mr. Rick Kalke for playing a vital role in the collection and analysis of the benthic biomass data. Mr. Kalke also supervised many technicians and graduate and undergraduate students over the years including Mary Conley, Landon Ward, Antonio Mannino, Chris Martin, Rob Rewolinsky, Amy Rutter and Greg Street. The authors thank Ms. Carol Simanek for playing a vital role in data management

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

1. INTRODUCTION

1.1. Background

The macrobenthos in Texas estuaries are affected by freshwater inflow (Kalke and Montagna, 1991; 1992). The freshwater inflows may be directly affecting the respiration, excretion, reproduction, natural mortality, and migration of macrobenthos through the variation of salinity (Montagna et al., 1995). It may be also indirectly affecting the consumption and assimilation of organic matter by macrobenthos if freshwater affects the primary production through the variation of nutrient input to each bay system. However, freshwater inflow is not the only important environmental factor regulating estuarine benthic subsystem. Predation by bottom fish, temperature change, light intensity, levels of particulate organic matter (POM), autotrophic production, and microbial production are other direct or indirect driving forces. A model is a way to combine all possible processes and environmental factors to gain insight into benthic dynamics, and explore the relationship between freshwater effects and other environmental effects. The quantitative modeling will also support the information needs that meet the Water Development Board's research program.

A major goal of The Water Development Board's research program is to define quantitative relationships between marine resource populations and freshwater inflows to the State's bays and estuaries. However, we know that there is year-to-year variability in benthic population densities and successional events of estuarine communities. This year-to-year variability confounds studies of different estuaries in different years. As a result of the Board funded research programs, we have demonstrated that long-term hydrological cycles regulate benthic abundance, productivity, and community structure in Texas estuaries. The high-inflow conditions during the period between 1992-1993 are very similar to the conditions we saw in 1987. Surprisingly, the wet period lasted almost 3 years, being one of the longest El Niño periods on record. We currently have data for 1½ climatic cycles: wet→dry→wet. This data can be used to test the hypothesis that long-term climate variability affects estuarine productivity.

A second major goal of the Texas Water Development Board is to create nitrogen budgets for all Texas bays and estuaries. Nitrogen is the key element that controls estuarine primary productivity. A simple budget would account for nitrogen entering the bay via freshwater inflow, how it is captured and transformed into biomass, and finally, how it is lost to the system. One aspect of nitrogen loss is very poorly understood. How much nitrogen is buried and lost from the system? This data gap is filled by taking deep sediment cores and measuring nitrogen changes with respect to sediment depth.

A third major goal of the Texas Water Development Board is to develop functional relationships between freshwater inflow and estuarine productivity. Populations of estuarine benthic infauna are affected by freshwater inflow in Texas estuaries (Montagna and Kalke, 1992). They appear to exhibit long-term variation in response to estuarine salinity changes, which are a function of historical climatic patterns. Enhanced productivity of benthic fauna seems to accompany a decrease of diversity in the Texas Estuaries. The temporal variation of benthic infauna may be due to multiple interactive effects of several environmental factors correlated with freshwater inflow, e.g., salinity, temperature, nutrients, and possibly pollution from land. Simple, multivariate statistical methods can be used to discover these relationships, but can not explain the functional dependencies of these effects. A modeling study based on a bio-energetic approach can be used to assess the role of freshwater inflow in controlling the temporal variability of benthic faunal functional responses, to freshwater inflow and benthic productivity.

Models are orderly and logical representations of underlying relationships; they reduce ambiguity and describe complexity with maximum parsimony. The use of models in ecology is useful, because of the inherent complexity of ecological relationships, the characteristic variability in ecological systems, and the apparently unpredictable effects of deliberate modification of the systems by man. It is difficult to understand benthic dynamics from a static modeling analysis (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 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.

1.2. Conceptual Model

Modeling of an ecosystem can start from a qualitative conceptual model. A conceptual model of South Texas estuaries has been created (Montagna et al., 1995). The purpose of this project was to place all of the perceived problems in the estuary into the context of natural ecosystem processes to support development of an estuarine management plan. This report was written at two levels: for the lay person and technical professional. It describes mechanistic relationships among biotic, abiotic, and anthropogenic compartments of the estuarine ecosystems contained within the Corpus Christi Bay National Estuary Program (CCBNEP) study area.

Previous studies indicate that it is easier to demonstrate freshwater effects by comparing estuaries with different freshwater inflow regimes than to study the salinity gradient within one estuary (Montagna and Kalke, 1992). This indicates that a synoptic approach is needed where several estuaries are studied simultaneously to establish the biological effects of freshwater inflow. The Lavaca-Colorado and Guadalupe Estuaries provide an interesting comparison. They receive about the same amount of inflow, but San Antonio Bay is smaller than Lavaca and Matagorda Bays and does not have significant, direct Gulf exchange; therefore salinity is much lower in the Guadalupe Estuary. Nueces Estuary and the Laguna Madre Estuary, which includes Baffin Bay, both have much lower river inflow volumes, and Nueces has direct exchange, but Upper Laguna Madre does not. Most importantly, by replicating estuaries, two with high inflow and two with low inflow, we can illustrate a general phenomenon.

Texas estuaries are different, because of differences in freshwater inflow. For example, the benthic macrofaunal community in the Guadalupe Estuary has a higher biomass and greater temporal variability than in the Nueces estuary (Montagna and Kalke, 1992). The differences in biomass levels and long-term variation of benthic infauna between the two estuaries may be due to several factors. There is a different amount of rainfall and inflow (Larkin and Bomar, 1983), different predator (i.e., finfish) stocks (Quast et al., 1988), and different amounts of primary production that may be transformed into food sources for benthic fauna (Montagna and Yoon, 1991). Polychaetes and mollusks are the two dominant taxa in both estuaries. They compose 91-97 % of total abundance and 91-98 % of total biomass (Montagna and Kalke, 1992). The composition of polychaetes and mollusks are different between the two estuaries. In the Guadalupe Estuary, mollusks (56 % in abundance and 80 % in biomass) are more dominant than polychaetes (41 % in abundance and 18 % in biomass). This trend is reversed in the Nueces Estuary, where polychaetes (78 % in abundance and 57 % in biomass) are more dominant than mollusks (13 % in abundance and 34 % in biomass). In general, mollusks are dominant in sandy sediments and feed on epibenthic diatoms and other food sources derived from primary production. In contrast, polychaetes are dominant in muddy sediments and feed on deposited POM. The differences between the estuaries suggests the hypothesis that there are two different trophic systems, one that has a food web based on primary production (and should dominate in the Guadalupe Estuary), and another that has a food web based on detritus or deposited POM (and should dominate in the Nueces Estuary). This hypothesis can be tested using a model of freshwater inflow influences upon the benthic faunal population in different estuarine ecosystems.

We have a continuous cycle of drought and flood conditions, which can greatly influence Texas Estuaries and water management decisions. These cycles regulate freshwater inflow, and thus, directly affect the biological communities. The variability in freshwater inflow cycle results in predictable changes in the estuary, which are diagramed in a conceptual model of the temporal sequence (Fig. 1.1)

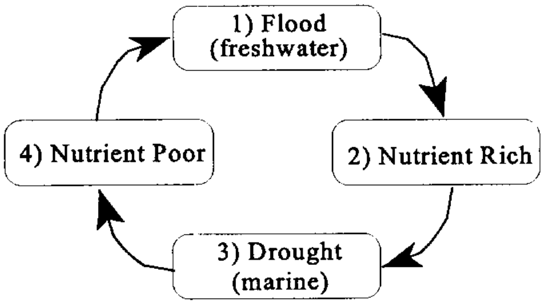


Fig. 1.1. Conceptual model of the biological effects of freshwater inflow.

Past studies in Texas estuaries demonstrate the biological effects of this cycle (Kalke and Montagna, 1991; 1992; 1995). Flood conditions introduce nutrient rich waters into the estuary that results in lower salinity. This usually happens very rapidly. During these periods the spatial extent of the freshwater fauna is increased. The estuarine fauna may even replace the marine fauna. The high level of nutrients stimulates a burst of benthic productivity of predominantly freshwater and estuarine communities. This is followed by a transition to low inflow conditions resulting in higher salinities, lower nutrients, marine fauna, decreased productivity and densities, and drought conditions. At first, the marine fauna may respond with a burst of productivity as the remaining nutrients are utilized, but eventually nutrients are depleted resulting in lower densities. The cycle is repeated with flooding and high freshwater inflows. This model represents a bio-mechanistic explanation of how benthos responds to freshwater inflow, and why their populations fluctuate from year-to-year as inflow varies.

1.3. South Texas Estuaries

The four estuaries that were studied in South Texas are the Lavaca-Colorado, Guadalupe, Nueces and Laguna Madre Estuaries. Two major water projects were initiated during recent years. The purpose of both projects was to increase freshwater inflows to bays in order to enhance secondary productivity. In 1990, the Texas Water Commission ordered The City of

Corpus Christi to release 151,000 ac-ft/y ($1.86\times10^8\,m^3\cdot y^{-1}$) to the Nueces Estuary from the Choke Canyon/Lake Corpus Christi reservoir system. The releases were mandated, because The City had not been releasing water. In the other project, the Colorado River was diverted into the eastern arm of Matagorda Bay by the creation of a flood diversion channel in 1991 and a dam in the river channel below the point of diversion in 1992. This project has diverted Colorado River water from the Gulf of Mexico into the eastern arm of Matagorda Bay.

Differences in biological processes among the four estuaries is driven by two dominant abiotic factors: freshwater inflow balance and physiography. Freshwater inflow drives many key abiotic ecological components, such as salinity, detritus, nutrients, and oxygen. Therefore, differences in freshwater inflow will have a great effect upon the structure and function of each system. The physiographic size and shape of the estuaries is also important in determining the dilution volume, residence time, currents, and exchange with the Gulf of Mexico. The four estuaries are also linked together by lagoons. Laguna Madre links Baffin Bay and Corpus Christi Bay. Red Fish Bay is actually a lagoonal system that links Corpus Christi Bay and Aransas Bay. Mesquite Bay links Aransas Bay to San Antonio Bay; and Espiritu Santo Bay links San Antonio Bay to Matagorda Bay). The intracoastal waterway flows through Lower Laguna Madre to Matagorda Bay increasing circulation in all systems. Bays are dominated by deeper, muddy sediments. Lagoons are shallower, narrower, with less fetch, have clearer water and more seagrass beds. These habitat differences cause the ecological differences among the four systems. Where as the Lavaca-Colorado, Guadalupe, and Nueces Estuaries are dominated by open bay, soft bottom habitat, Laguna Madre is dominated by submerged vegetated habitat. Another difference among estuaries is the size and volume of Gulf water exchanged via passes, which are conduits for fisheries recruitment, and the exchange of nutrients and organic matter with the Gulf of Mexico. The same basic habitats and processes occur in all Texas estuaries.

1.3.1. Lavaca-Colorado Estuary

Lavaca-Colorado Estuary is the largest estuary among the four Texas estuaries studied (Table 1.1). It is about 1,158 km^2 in size and has the largest freshwater inflows, 3,242 $10^6\ m^3\ yr^{-1}$ which is 1.5 times that of the Guadalupe Estuary, and 6 times that of the Nueces Estuary. The freshwater inflows are relatively constant, and the bottom salinity has only little variation within a year. The phytoplankton abundance is less than in the Guadalupe estuary, but 4 times higher than in the Nueces and Laguna Madre estuaries. The zooplankton abundance is about same as in the Guadalupe Estuary, or 1,000 times less than in the Nueces and Laguna Madre Estuaries. The density of the benthos community, such as mollusks, is lower than all other estuaries.

Table 1.1. Comparison of physical characteristics of four Texas estuaries.

Characters	Estuary			
	Lavaca-Colorado	Guadalupe	Nueces	Laguna Madre
Size[1] (km^2)	1,158	551	433	1,139
Rainfall[1] average ($cm\ yr^{-1}$)	102	91	76	69
Inflow average ($10^6\ m^3\ yr^{-1}$)	3,242	2,545	509	-947
Bottom salinity[1] average (ppt)	22-22.2	13-18.6	16.6-30.6	31.3-37
Maximal monthly mean inflow[1] ($m^3\ s^{-1}$)	140 (1938-1990)	120 (1939-1989)	50 (1939-1989)	5 (1965-1987)
Average depth (m)	2	1	2	1
Maximal phytoplankton abundance[2] (10^{-3}cell ml^{-1})	4.5	19	1.1	1.6
Maximal zooplankton abundance[2] (10^5 ind. m^{-3})	0.3	0.5	500	200
Maximal benthos abundance[3] (ind. m^{-2})	9,700	35,000	7,200	13,000

[1]Orlando et al. (1993); [2]summarized by Armstrong (1985), Montagna et al. (1995); [3]Montagna & Kalke (1995), mollusks only.

1.3.2. Guadalupe Estuary

The Guadalupe Estuary is relatively small (551 km^2) and has high inflow rates (averaging 2,545 10^6 m^3 yr^{-1}). This results in the lowest average salinities among the four estuaries studied. The Guadalupe Estuary has the highest phytoplankton abundance in comparison with other estuaries. The average density is 19,000 cells ml^{-1}, which is 4-20 times higher than in other estuaries. Benthos abundance, such as mollusks, is 3-30 times higher than in other estuaries. The high mollusk density is probably due to high inflow into a small estuary, which results in low salinities that mollusks require, and the high densities of phytoplankton, which are the main food source for mollusks.

1.3.3. Nueces Estuary

Nueces Estuary has a higher level of nutrient storage, kinetic storage and oxygen storage in comparison with Laguna Madre Estuary, because of more inflow from rivers and creeks per unit time. However, phytoplankton primary production in Corpus Christi Bay is only 0.48-1.26 g C m^{-2} d^{-1} (Odum and Wilson, 1962; Odum et al., 1963; Flint, 1984; Stockwell, 1989). Other limitations rather than nutrient limitations are more important in this estuary. The average water depth in the Nueces Estuary is 2 m (compared to 1 m for Laguna Madre). The total area is relatively small (433 km^2), which is about half or one-third that of Laguna Madre. The amount of sun radiation per unit water volume is much lower than in Laguna Madre. This means a lower capacity for temperature storage and lower ecological efficiency than in Laguna Madre. This may explain the lower primary production rates and lower amount of consumer biomass.

The Nueces Estuary has both river and creek inflow sources while Laguna Madre has only creeks. The Nueces Estuary has an annual inflow balance ($509 \cdot 10^6\ m^3 \cdot y^{-1}$) that is much higher than in the Laguna Madre. This indicates that energy flow from the land to the bay system is higher than in the Laguna Madre Estuary. However, the energy flow from the bay area system to ocean is also higher. The average rainfall of the four estuaries decreases from Nueces ($76\ cm \cdot y^{-1}$) to Laguna Madre ($69\ cm \cdot y^{-1}$). The residence time of water in the Nueces Estuary is 0.46 y. The gradient of river inflow and average rainfall cause a lower salinity in the Nueces Estuary (17-31‰) than in Laguna Madre Estuary (30-37‰). High salinity variation makes this ecosystem more unstable than Laguna Madre, by affecting population aging rates, respiration rates, reproduction rates and migration rates (Montagna et al., 1995). The Nueces Estuary is also more stressed, because there are more people living on their shores.

1.3.4. Laguna Madre

Laguna Madre is as large (1,139 km^2) as the Lavaca-Matagorda Estuary. Laguna Madre has an average depth that is shallower than the other estuaries and a larger surface area receiving sun light (Montagna et al., 1995). Therefore, the sun radiation per unit water volume is much higher in Laguna Madre than in the other estuaries, so the temperature storage is higher. This increases the ratio of energy flow between subsystem and storage components, by increasing all biophysiological process, such as synthesis rate, intake rate, decomposition rate, aging rate, respiration rate, migration rate, and reproduction rate (Montagna et al., 1995). The high energy flow in this ecosystem creates the high rate of primary production. Phytoplankton primary production in Laguna Madre ranges from 2.68 to 4.78 $g\ C \cdot m^{-2} \cdot d^{-1}$, which is more than two times that of the Nueces Estuary (Odum and Wilson, 1962; Odum et al., 1963). The high ecological efficiency also results in the high abundances of the higher level consumers, such as benthic mollusks, and fishes (Montagna et al., 1995). The benthic mollusk abundance in Laguna Madre (13,000 $ind. \cdot m^{-2}$) is twice that of the other estuaries (2,500-7,200 $ind. \cdot m^{-2}$). The commercial harvest of finfish in Laguna Madre (834 $10^3\ kg \cdot y^{-1}$) is about four times higher than the others (151-207 $10^3\ kg \cdot y^{-1}$). This biomass productivity is probably due to overall higher primary production in Laguna Madre. Laguna Madre Estuary has a lower energy input from rivers, which provide nutrients, in comparison with the other estuaries. Laguna Madre has a negative inflow balance ($-947 \cdot 10^6\ m^3 \cdot y^{-1}$), which means the freshwater inflow is less than outflow, e.g., evaporation. The negative balance also accounts for hyper salinity in Laguna Madre. Flow of water from Gulf passes and the Nueces Estuary keeps the Laguna from evaporating entirely. Residence time for the water in Laguna, however, is very difficult to calculate because of its shallow depth, "negative" inflow, and its connection to Nueces Estuary (Montagna et al., 1995). Energy flow from ocean to the bay area system is higher for Laguna Madre than for the others (Montagna et al., 1995), which mainly provides detritus. The detritus storage tank in Laguna Madre is expected to be much higher than in other estuaries, because there is high primary production due to an extensive seagrass habitat. The consumer subsystem is dominated by deposit feeding benthos. The input of seawater and less input from rivers also maintains a stable high salinity, but low nutrients in Laguna Madre. The main limitation on producers' synthesis may be only nutrients (Montagna et al., 1995). However, less input from inflow reduces the anthropogenic effects. So, Laguna Madre Estuary has a higher temperature storage, salinity storage and detritus storage and remains a more natural ecosystem than others.

1.4. Goals

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 of criteria. The conceptual model has high generality for Texas estuarine systems, but low realism and low precision. A quantitative model is required to provide the realism and precision, and test hypotheses from the conceptual model.

A quantitative model requires a long-term data set to calibrate the model. We have long-term macrobenthos data for four Texas Estuaries: Lavaca-Colorado Estuary, Guadalupe Estuary, Nueces Estuary and Laguna Madre Estuary (which includes Baffin Bay). Each estuary can be subdivided into a primary and secondary bay with different levels of freshwater effects, due to the gradient from river to sea. Comparison between the two bays within each estuary also represents affects of freshwater. It is possible to model those four estuaries as eight separate bays. In this way we test inflow affects in two ways: within estuaries and among estuaries.

In summary, the objectives of this quantitative modeling study include:

1) To test the heuristic model describing the cyclical nature of inflow and its consequent biotic effects.
2) To describe the successional events during various phases of the hydrological cycles.
3) To develop quantitative relationships between freshwater inflow and benthic abundances.
4) To determine the role of Gulf exchange (marine tidal flow) versus freshwater inflow on benthic communities in Texas bays and estuaries.
5) To determine the role of inter-annual variability in modifying benthic productivity.
6) To develop a model to predict annual production and production efficiency of benthic infauna, based estimates of standing stocks in four Texas Estuaries.
7) To determine the role of non-freshwater inflow effects, such as, fish predation.

2. MATERIALS AND METHODS

2.1 Study Sites

Four to six stations were chosen in four estuaries (Fig. 2.1., Table 2.1). Generally, two replicate stations (A and B) are in the secondary bay where freshwater influences are greatest, and two other replicate stations C and D) are in the primary bay where marine influences are greatest. By using two stations in the freshwater influenced zone and two stations in the marine influenced zone we are replicating effects at the treatment level and avoiding pseudoreplication. There has been a diversion of the Colorado River into the east arm of Matagorda Bay, so we have located two additional stations (E and F) there. The stations in the Laguna Madre Estuary are located using the paired-station strategy. Two stations are located in Baffin Bay (6 and 24), and two stations are located in Laguna Madre: in a seagrass bed (189G) and an unvegetated sand patch (189S). The station data for each bay was pooled, so eight major bays were characterized.

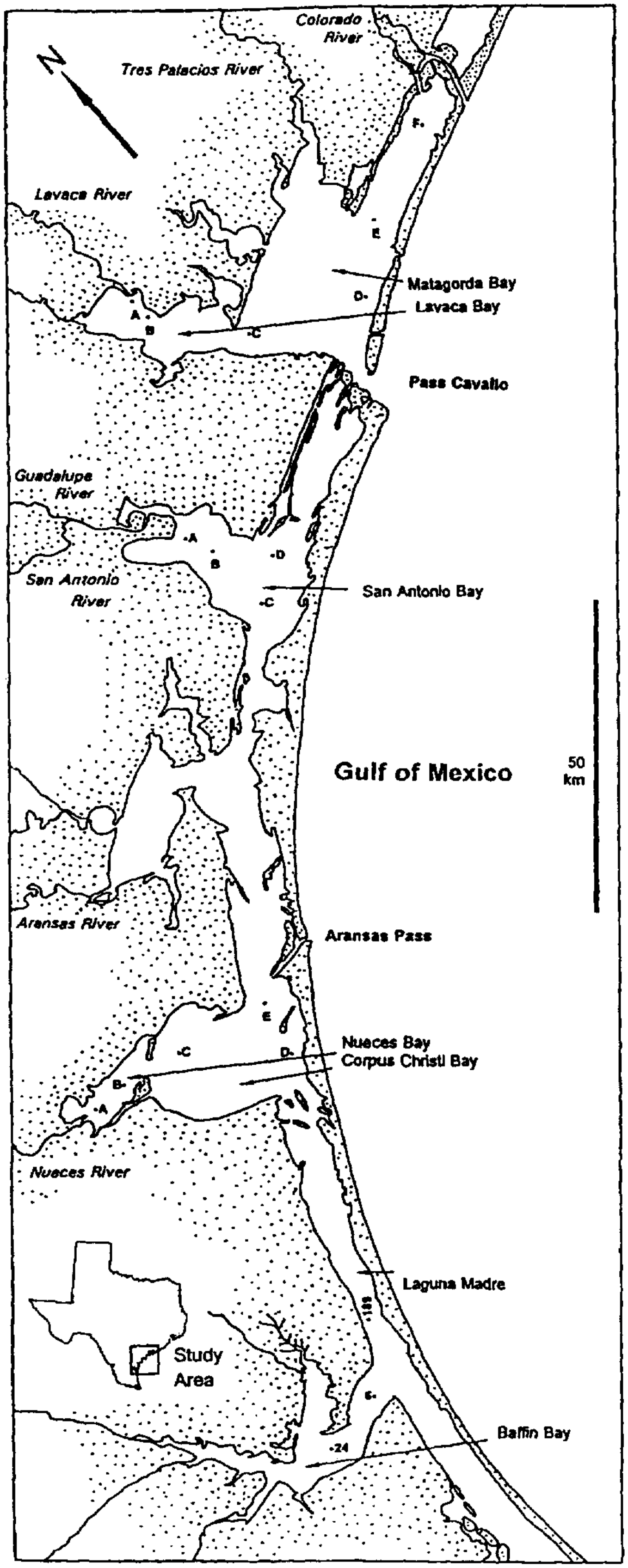


Fig. 2.1. The study area and station locations.

Table 2.1. Location of sampling stations and sampling periods with four estuaries. Table gives estuary name, bay type, bay name, stations and years of the sampling.

Estuary	Bay Type	Bay Name	Station	Sampling Period
Lavaca-Colorado	Secondary	Lavaca	A	1984-1995
	Secondary	Lavaca	B	1988-1995
	Primary	Matagorda	C, D	1988-1995
	Lagoon	East Matagorda	E, F	1993-1995
Guadalupe	Secondary	Upper San Antonio	A, B	1987-1995
	Primary	Lower San Antonio	C, D	1987-1995
Nueces	Secondary	Nueces	A, B	1988-1995
	Primary	Corpus Christi	C, D	1988-1995
	Primary	Corpus Christi	E	1988-1995
Laguna Madre	Secondary	Baffin Bay	6, 24	1988-1995
	Primary	Laguna Madre	189G, 189S	1988-1995

2.2. Modeling Procedure

The model consisted of several mathematical equations that calculated the variation of benthos biomass in response to the variation in environmental data. Model input was the observed long-term environmental data, and output was simulated benthos biomass over time. When the model input is fixed, simulation of observations can be improved by changing the mathematical equations. Changing the equations form is process of model development. Changing parameter values in the equations to improve the simulation of observations is model calibration. The calibration is re-done each time a new data or model structure is set up. The simulation of the observations is based on the calibrated parameters. So, to model an ecosystem, several repeated cycles of structure, calibration, and simulation are required until the simulation of observations is satisfactory (Fig. 2.2). Each time, sensitivity analysis is performed before the calibration to judge if the output range of the simulation will cover the range of all observations. A parameter should be sensitive enough to change the model output, if not the parameter can be deleted.

Assembling a database is the first step. It includes creation of databases for salinity, biomass, and other variables. The data may be used as either observations for model calibration, input to the model, or as a forcing function to drive the model. Statistical analyses can be used to determine significant environmental factors, to simplify the model by reducing the unimportant variables in the model. The data collection is also used to setup initial parameter ranges. It is efficient when the initial parameter ranges are as narrow as possible, because it reduces calibration effort, and limits the possibility of a wrong calibration direction. However, when the ranges are unknown, the range of parameter values must be as large as possible to include all possibilities. Previous modeling studies are the best information for developing the model structure, calibration, and validation techniques.

The model structure is the main step in the procedure. It requires several revisions. Every time a new model is setup, thousands of calibration runs are required to estimate the parameters. The simulation of the data is based on the calibrated parameters and can be used to perform model validation. Redesign of the model structure is required when the simulation and validation are unsatisfactory. Every change in the model structure will be followed by another sensitivity, calibration, simulation and validation step (Fig. 2.2).

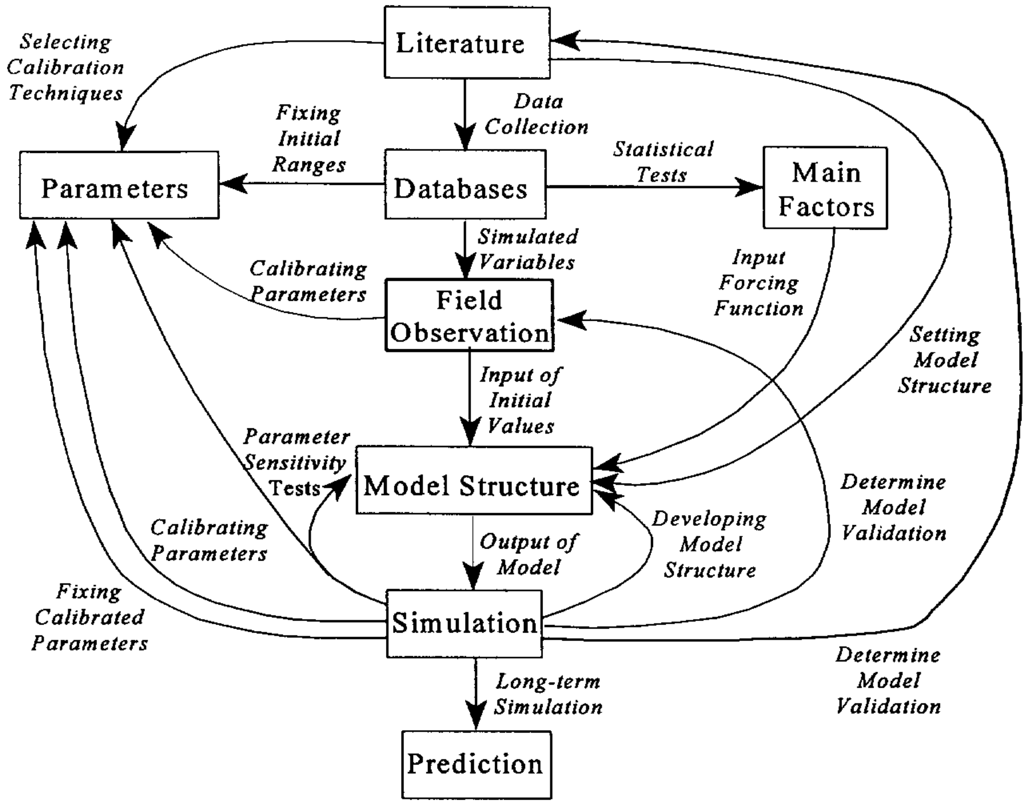


Fig. 2.2. Flow diagram of the steps used to develop the model.

2.3. Databases

The availability of a database is the most important component for a modeling study. It can determine the success of a modeling project. A good database provides a high number of observations, information from the literature, and low error rates due to high sample numbers. These characteristics of the database ensure the correct determination of parameters.

2.3.1. Benthos

We began sampling the Lavaca-Colorado Estuary in November 1984 (Kalke and Montagna, 1991). The Guadalupe Estuary was sampled bimonthly during 1987 (Montagna and Kalke, 1992). From these two studies, we learned that long-term changes in benthos within these estuaries could be characterized by sampling four stations, four times per year. Starting in July 1988, we began a sampling program to compare the Lavaca-Colorado and Guadalupe Estuaries (Table 2.2). The Board funded the first year (1988) of the program. Since then, we have expanded the program to the eastern arm of Matagorda Bay with funding from the Lower Colorado River Authority. The goal for establishing these stations was to assess the effect of the Colorado River diversion on estuarine productivity. The Nueces Estuary was sampled in 1987 on a Board funded project (Montagna and Kalke, 1992), but since then, sampling has been performed under other projects funded by the Coastal Bend Bays Foundation, Texas A&M Sea Grant Program, Corpus Christi Bay National Estuary Program, or self funded. The Laguna Madre and Baffin Bay sampling has been funded by the Texas Advancement Technology Program. Combining these sources of funding, we have assembled a long-term, coast-wide database on benthic biomass, abundance, and community structure (Table 2.2).

During each sampling event, hydrographic measurements are also made, which includes nutrient concentrations, salinity, temperature, and water depth. Once each year, usually in October, sediment grain size, total nitrogen, and organic carbon content are measured in sediments.

Table 2.2. Years in which benthic data is available. Samples taken per year in four the Texas estuaries. (LC= Lavaca-Colorado Estuary, GE=Guadalupe Estuary, NE=Nueces Estuary, LM=Laguna Madre. For macrofauna abundance, biomass and community data: X=samples taken four time a year (mostly in January, April, August and November), M=samples taken more than four times a year, L=samples taken less than four times a year, and -=not sampled).

Estuaries	Sampling Years (1987-1995)								
	87	88	89	90	91	92	93	94	95
LC	-	L	L	L	X	X	X	X	X
GE	X	L	L	L	X	X	X	X	X
NE	L	X	-	L	X	X	X	X	X
LM	-	L	X	X	M	X	X	X	X

2.3.2. Predators

Fisheries data were obtained from Texas Park and Wildlife Department (McEachron and Fuls, 1996). The Coastal Fisheries Division samples monthly in the four estuaries using a shrimp trawl and bag seine. The data are available from 1988 to present. In a study of mercury bioaccumulation in different food chains, Montagna and Kathmann (in prep.) determined that black drum, red drum and blue crab are the main predators on benthic infauna. We used the average value for these three main predators from shrimp trawls for all stations within a bay.

2.3.3. Other Environmental Data

We record salinity, temperature, and water depth at the same time benthos is sampled. Nutrient data for many of the same stations and periods are available (Whitledge, 1996). However, there is not enough data to form a time series for primary production. A range of values for primary production from previous studies are available (Table 2.3). Monthly day-length for the Texas coastal was obtained from Tony Amos at UT Marine Science Institute. Table 2.4. summarizes the data in the continuous long-term database assembled for modeling Texas estuaries.

Table 2.3. Available data for primary production from previous studies.

Bay	Previous record of primary production (g C $\cdot$ m^{-2} $\cdot$ d^{-1})	References
Lavaca	0.5 - 2.4	Brock (1994)
Matagorda	0.5 - 2.4	Brock (1994)
Upper San Antonio	0.3 - 1.8	Stockwell (1989)
Lower San Antonio	0.5 - 3.85	Stockwell (1989)
Nueces	0.35 - 1.7	Stockwell (1989)
Corpus Christi	0.75 - 4.1	Stockwell (1989)
Baffin	1.2 - 4.1	Odum & Wilson (1962)
Upper Laguna Madre	2.75	Odum et al. (1974)

Table 2.4. Summary of the data variables assembled for the continuous long-term database for modeling four Texas estuaries. Period is given as the year and month for each estuary. Abbreviations as in Table 2.2.

Variable	Estuary			
	LC	GE	NE	LM
Temperature	1988.04-1996.04	1987.01-1995.10	1987.10-1996.01	1988.04-1996.01
Salinity	1988.01-1996.01	1987.01-1995.10	1987.10-1996.01	1988.04-1996.01
Water depth	1988.04-1996.01	1987.01-1995.10	1987.10-1996.01	1988.04-1996.01
Nutrients (N, P, Si)	1991.10-1995.10	1987.01-1988.07, 1991.04-1995.10	1987.03-1988.07, 1991.04-1995.10	1991.10-1996.01
Fish	1987.01-1996.04	1987.01-1996.04	1987.01-1996.04	1988.01-1996.04
Benthos	1988.04-1995.10	1987.01-1995.10	1987.10-1995.04	1989.03-1994.10

2.4. Model Structure

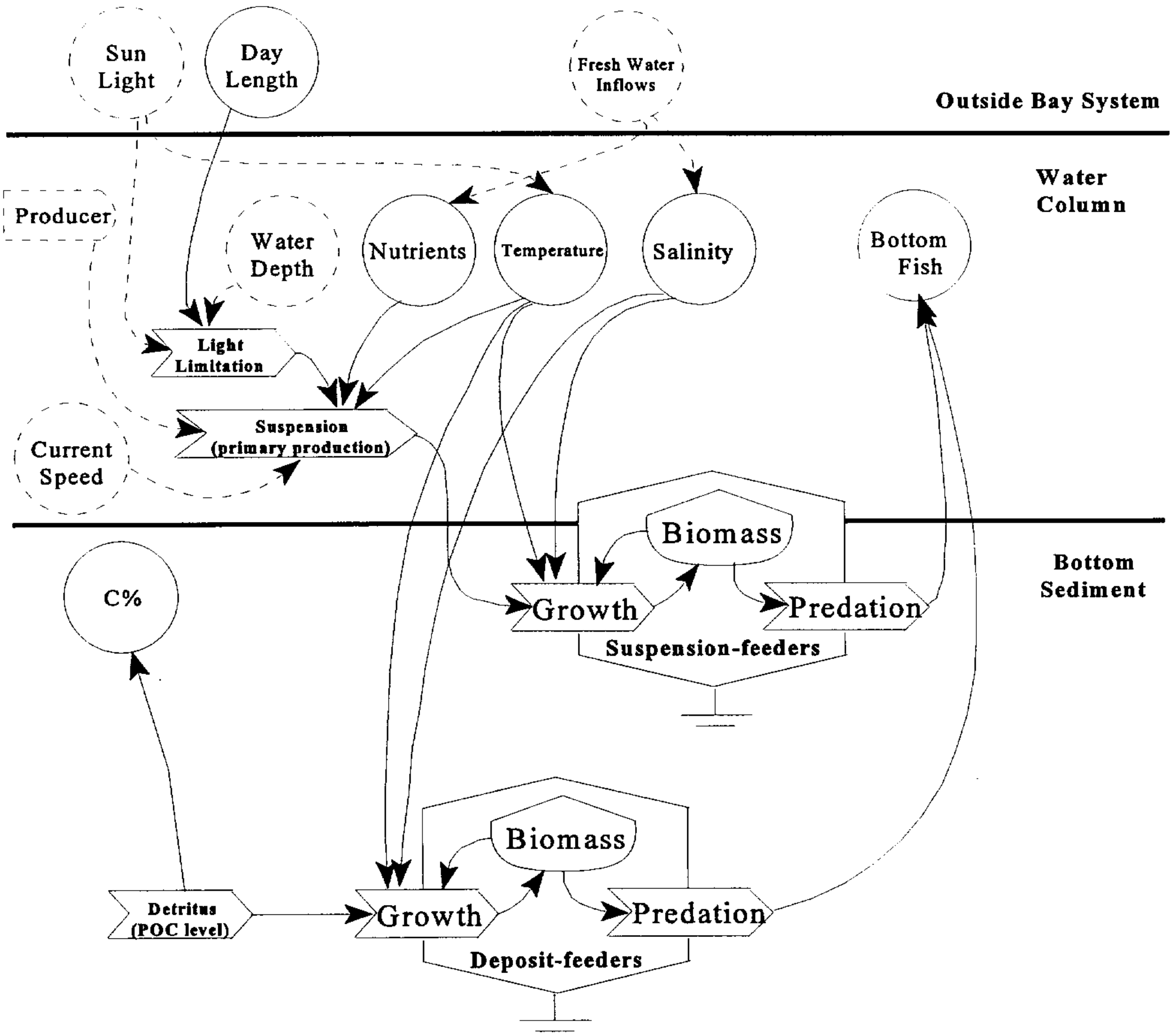
The long-term, benthic infauna, data set from Texas estuaries was used to calibrate a temporally dynamic model of biological processes. The relationship between biomass of benthic infauna and environmental factors associated with freshwater inflow is incorporated into the model. The macroinfauna was divided into two trophic groups. Infaunal suspension feeders and deposit feeders were modeled separately. To test for inflow effects, the ideal input to the model would be freshwater inflow as the basic forcing function. However, inflow rates have variable effects depending on the physiographic characteristics of each estuary. Therefore, a physical model that predicts salinity change would be needed to provide input to the biological model. To avoid this level of complexity, the empirical salinity values were used as input. In this way, salinity is used as a surrogate for inflow. Salinity values represent the integration of all the physical characteristic of the estuary, e.g., size, inflow, outflow, Gulf exchange, and climatic variability. Other inputs to the model included fish and crabs as predators, temperature, and environmental data that affects primary production (e.g., day length and nutrient concentrations).

Odum (1971, 1972 and 1983) energy circuit language is used to present the model for simulating the Texas benthos biomass (Fig.2.3). The current model includes four forcing functions: salinity, temperature, food sources and predators. They drive the model mainly through four environmental limitations: salinity limitation, temperature limitation, food availability limitation, and predation limitation. Other forcing functions (e.g., nutrient concentrations, day length, and water depth) drive the model through the estimation of food source availability by the calculation of primary production.

There are two main trophic guilds in the benthos: the grazing food-chain and the detrital food chain. Grazers utilize autotrophic, production and detritivores utilize heterotrophic production. To simplify the model, all macrobenthos were separated into one of two groups: the suspension feeders and deposit feeders. Suspension feeders are defined as those benthos who obtain their food sources through 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. 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. Combining all these taxa into just two groups is a simplification. Many macrobenthos, e.g., mollusks and polychaetes, can alternate between being suspension feeders and deposit feeders. However, this simplification allows us to define suspension feeders as benthos limited by autotrophic food sources, and deposit feeders as those organisms limited by heterotrophic food sources. Descriptions of the community structure of these estuaries have already been published or are in preparation (Kalke and Montagna, 1991; Montagna and Kalke, 1992, 1995; Martin and Montagna, 1995; Mannino and Montagna., 1996; Conley et al., 1997)

Modeling benthos secondary production is not as easy as modeling primary production. The benthic food web is complex and secondary production rates are not a function of physical-chemical variables. Primary production, which is growth based on sun radiation and nutrient concentrations, is the main food source for suspension feeders, that consume phytoplankton and benthic diatoms. Therefore it is necessary to predict the food sources for suspension feeders in a model. Deposit feeders primarily consume POM, and this can be approximated by the concentration of total organic carbon (TOC) in sediments, which is empirically derived. The accumulation of POM, and the variation of nutrient concentrations and salinity due to temporal variations of freshwater inflows are not simulated here. The measured concentration of nutrients and salinity are used as the input for the model. The mathematical formulae are based on known bioenergetic mechanisms of invertebrates.

Fig. 2.3. Energy circuit diagram for the structure of the benthic macrofauna biomass model Dashed lines are the parts not included in the model.



2.4. Development of Mathematical Formulae

The basic formula that describes the change of benthic biomass over time (Li et al., 1996) is based on the law of conservation of mass (Crisp, 1971), and in the form:

$$\frac{d(B)}{d(t)} = I \cdot A - L - D \tag{1}$$

Where B is the benthos biomass, t is time, I is the total intake of food by benthic infauna, A is average assimilation efficiency of benthic infauna, L is the total loss due to respiration, excretion and age related mortality, D is total mortality caused by predators. Unfortunately, it is not possible to simulate B terms of I and L, because observed data on food source standing stocks and respiration of benthos are rare and incomplete in the study area. Our approach is to replace net growth rate in place of I and L.

2.4.1. Growth Rate of Benthos Biomass

The net growth rate is used in place of the intake rate, assimilation efficiency, respiration rate, aging mortality and excretion rate. The formula becomes a Lotka-Volterra growth rate model (Lotka, 1925) in the form:

$$\frac{d(B)}{d(t)} = r \cdot B - g \cdot F \tag{2}$$

Where r is the net growth rate without predation pressure. The predation loss is calculated by feeding rate of predators, g, and the density of predatory fish, F.

A logistic limitation term to growth rate is suggested by Brown and Rothery (1993), it takes the form:

$$\frac{d(B)}{d(t)} = r \cdot B \cdot \left(1 - \frac{B}{c}\right) - g \cdot F \tag{3}$$

Where c is the biomass carrying capacity for a population that is limited by space. The c in equation (3) is only a limitation for the capacity of biomass. The limitation of a population and its biomass is also due to many other environmental effects. Our model is based on equation (3), and has been modified to include environmental limitation. The new equation contains a parameter to reduce the maximal growth rate (r) and maximal predation rate (g) by the effects of environmental limitation (E). The values of E are between 0 and 1. When E=1, there is no environmental limitation, and the benthic population reaches maximal growth rate or predators reach maximal feeding rate. When E=0, environmental factors reach maximal limitation, benthic populations do not grow, or predators do not consume benthos. As there are also more than one predator, the final equation for the model becomes:

$$\frac{d(B_{(i,j)})}{d(t)} = r_{(i)} \cdot E_{ben(i,j)} \cdot B_{(i,j)} \cdot \left(1 - \frac{B_{(i,j)}}{c_{(i)}}\right) - \sum_{k} g_{(i,j,k)} \cdot E_{fish(i,j)} \cdot F_{(j,k)} \qquad \textbf{(4)}$$

Where i=1 or 2 for deposit feeders or suspension feeders; j=1-8 for eight bay systems; k=1-3 for three different predators: red drum, black drum and blue crab. The net growth rate is $r_{(i)}$. The environmental limitation for benthos biomass growth is $E_{ben(i,\,j)}$. The biomass carrying capacity levels for the two feeding groups is $c_{(i)}$. The predation rate by fish k in bay j to prey benthos i is $g_{(i,\,j,\,k)}$. The term $E_{fish(i,\,j)}$ is the environmental limitation for predation. The different benthos species have different biomasses in each bay, computed by their r, E, c and g. Predator abundances are also different in the eight bays. The benthos should have the same r and c in all eight bays. However, the dominant species in the deposit-feeding group and suspension feeding group are different in the different bays. So, it is necessary to run the model separately for each of the eight bays..

The term $E_{ben(i,\,j)}$ includes three effects: temperature limitation ($E_{tem(j)}$), salinity limitation ($E_{sal(i,\,j)}$), and food concentration limitation ($E_{food(i,\,j)}$):

$$E_{ben(i,j)} = E_{tem(j)} \cdot E_{sal(i,j)} \cdot E_{food(i,j)} \qquad \textbf{(5)}$$

2.4.2. Temperature Limitation

An exponential equation was used for the temperature effect (Carrada, 1983):

$$E_{tem(j)} = e^{\frac{T_{(j)} - T_{\max}}{p_{(1)}}} \qquad \textbf{(6)}$$

Where $E_{tem(j)}$ is the temperature limitation, T is the temperature, and T_{max} is the most suitable temperature, which is fixed at the highest temperature recorded at each location. When T is close to $p_{(1)}$, $E_{tem(j)} = 1$, and there is no temperature limitation. Therefore, $p_{(1)}$ is a parameter that describes the weighing due to temperature limitation. The higher $p_{(1)}$ is, the higher the sensitivity to temperature (Fig. 2.4).

2.4.3. Salinity Limitation

The salinity effects are the most interesting environmental factor. Salinity is directly correlated with freshwater inflow. All invertebrates have suitable salinity ranges, at which the population growth is maximal, i.e., the highest metabolism rates (Wohlschlag et al., 1977). An exponential equation is used to model salinity limitation. The equation is similar in form to that used for temperature limitation:

$$E_{sal(i,j)} = e^{\frac{S_{(j)} - p_{(i,3)}}{P_{(2)}}} \tag{7}$$

Where $E_{sal(i,j)}$ is the salinity limitation, $S_{(j)}$ is salinity, $p_{(i,3)}$ is the optimal salinity for a population, and $P_{(2)}$ is a parameter that describes the weight of the salinity limitation. There is no salinity effect when $P_{(2)}=\infty$. Salinity limitation has a centralized optimum, with greater affects at high and low salinities (Fig. 2.5). The greater the salinity tolerance range, the higher the $P_{(2)}$ value (Fig. 2.5).

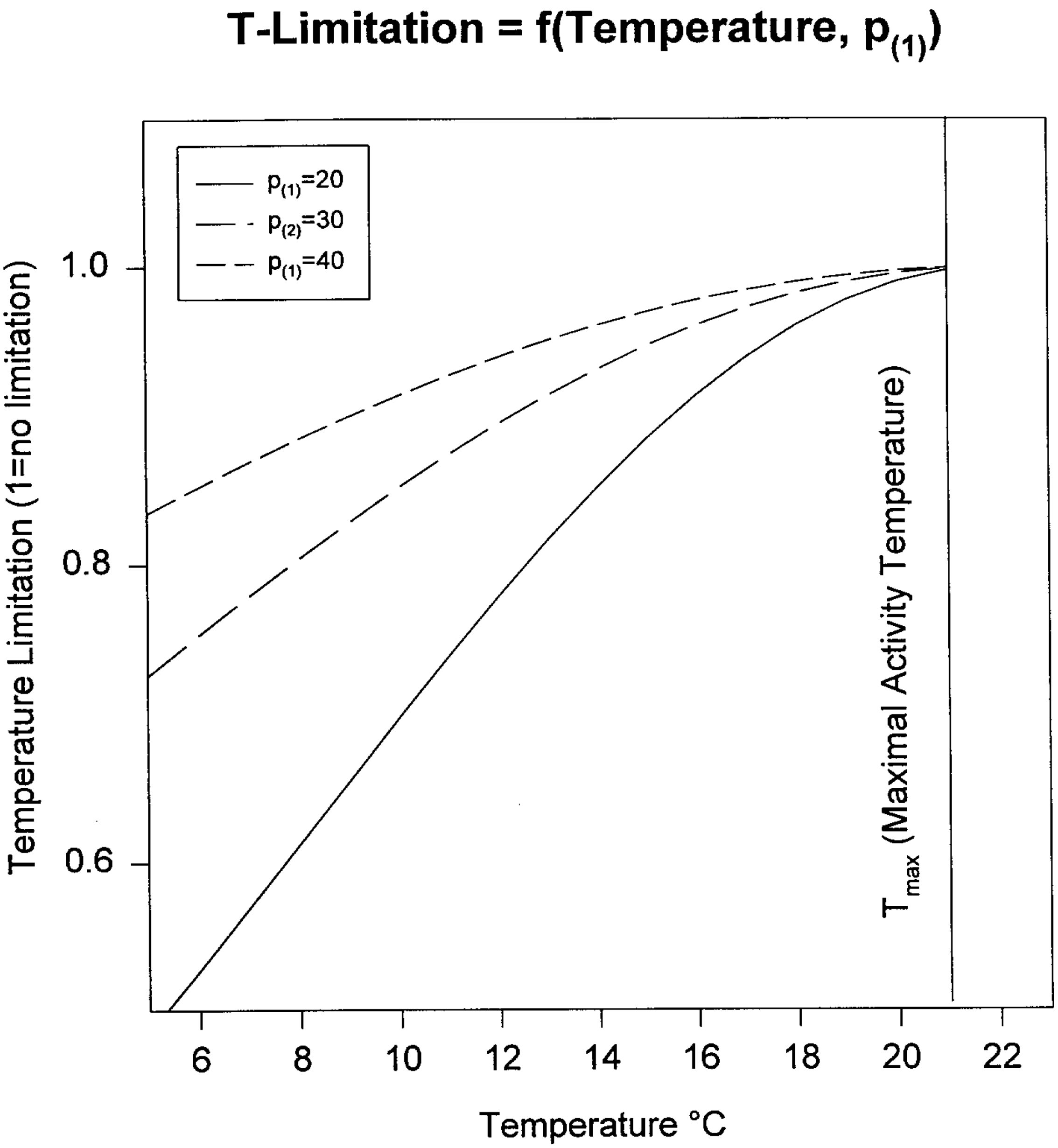


Fig. 2.4. The temperature limitation that formulated by equation (6).

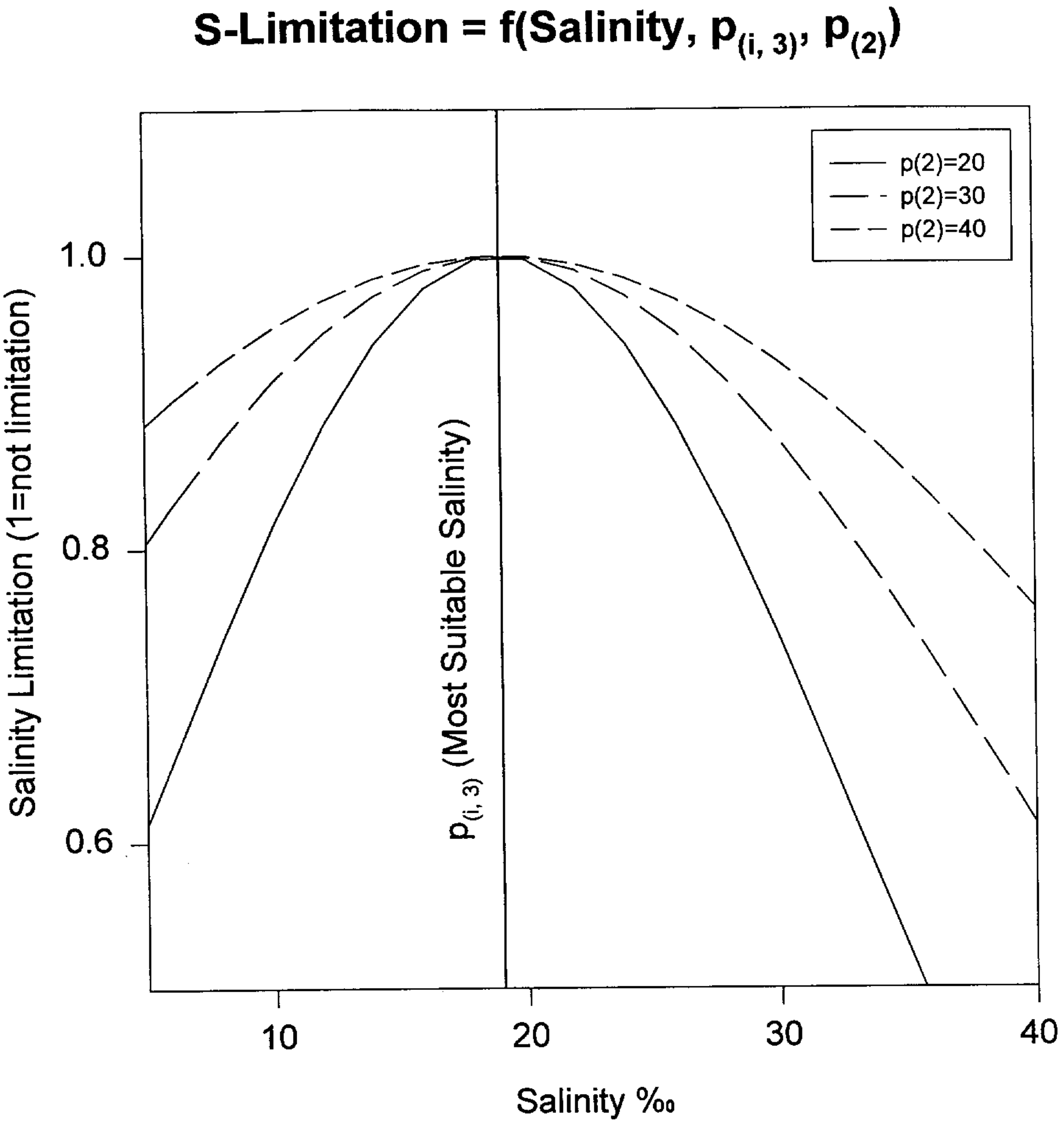


Fig. 2.5. The salinity limitation that formulated by equation (7).

2.4.4. Food Sources Limitation

Michaelis-Menten kinetics is used to describe the food source limitation (see a review, Keen and Spain, 1992):

$$E_{food(i,j)}=\frac{F_{(i,j)}}{F_{(i,j)}+p_{(i,4)}} \quad \textbf{(8)}$$

Where $E_{food(i,j)}$ is the food limitation, $F_{(i,j)}$ is the concentration of food source for the infauna, which is sedimentary POC for deposit feeders and primary production for suspension feeders, and $p_{(i,4)}$ is a parameter at which the food concentration is at half the maximum level of the population growth rate.

As two feeding groups were simulated (deposit feeders and suspension feeders), there were two different food sources: detritus in sediment and organic matter in the water. Sedimentary POM was used as food sources for deposit feeders, and expected primary production was used for suspension feeders. Increased consumer biomass ($B_{(i,j)}$) can increase food limitation. Therefore, Equation (8) transforms to the ratio as a function of benthic biomass:

$$E_{food(i,j)}=\frac{\frac{F_{(i,j)}}{B_{(i,j)}}}{\frac{F_{(i,j)}}{B_{(i,j)}}+p_{(i,4)}} \quad \textbf{(9)}$$

2.4.4.1. Food Sources for Deposit-feeders

The sedimentary POM is expected to be a constant level in each bay. We used eight parameters as POM levels, one for each of the eight bays. The POM levels were pre-calibrated by the observed carbon concentration ($C\%_{(j)}$) in the sediment (j=1-8 for eight bays):

$$C\%_{(j)}=\frac{p_{pom(j)}}{p_{sed}}\cdot 100\% \quad \textbf{(10)}$$

Where $p_{pom(j)}$ is the sedimentary POM level for each bay, and p_{sed} is a parameter for the average dry weight of whole sediment. The POM levels for each bay are the food sources for deposit-feeders ($F_{(i,j)}$) in each bay:

$$F_{(1,j)}=p_{pom(j)} \tag{11}$$

2.4.4.2. Food Sources for Suspension-feeders

Primary production is expected to be the main food source for suspension feeders. Primary production is simulated as a function of day length, temperature, nutrient concentration and water depth. Primary production was pre-calibrated using data from previous studies (Stockwell, 1989; Armstrong et al., 1985) using the following formula:

$$F_{(2,j)}=p_{mic(2)}\cdot p_{mic(3)}\cdot L_{(t)}\cdot E_{nut(j)} \tag{12}$$

Where $F_{(2j)}$ is the available food source for suspension feeders, $p_{mic(2)}$ is the maximal monthly primary production rate, $p_{mic(3)}$ is the temperature limitation for primary production, $L_{(t)}$ is the day length that represents light limitation, and $E_{nut(j,\ t)}$ is the nutrient limitation for photosynthesis that includes concentrations of nitrogen (N), silica (Si), and phosphorus (P).

4.4.4.3. Nutrient Limitation

Nutrient limitation ($E_{nut(j)}$) for photosynthesis was modeled according to the Redfield ratio of 106:16:15:1, which assumes that producers use carbon, N, Si, and P proportionally by weight (Redfield, 1934, Parsons et al., 1961):

$$E_{nut(j)}=MIN\left(\frac{\frac{[N]_{(j)}}{[N]_{(j)}+p_{mic(1)}}}{16},\frac{\frac{[P]_{(j)}}{[P]_{(j)}+p_{mic(1)}}}{1},\frac{\frac{[Si]_{(j)}}{[Si]_{(j)}+p_{mic(1)}}}{15}\right) \tag{13}$$

Where $[N]_{(j)}$, $[P]_{(j)}$, and $[Si]_{(j)}$ are concentrations of inorganic nitrogen, phosphorus and silica.

4.4.2.5. Day Length Limitation

Photosynthesis is limited by light, which varies seasonally. A sine function is used to simulate the seasonal cycle of day length:

$$L_{(t)}=p_{avg}+p_{amp}\cdot\sin\left(\frac{2\Pi\cdot(t)}{12}-p_{pha}\right) \tag{14}$$

where $L_{(t)}$ is day length at time t, p_{avg} is the average day length over a year, p_{amp} is the amplitude of the seasonal fluctuation, and p_{pha} is the correction factor for starting the phase of the sine cycle at a given time.

2.4.5. Prey Density Limitation

Predation can be limited by temperature, salinity, prey abundance, and predator density. In this study, we considered only the benthic prey abundance, because predation rates on benthos is strongly related to the benthos biomass. A complete ecosystem model would also include fish bioenergetics. Predator limitation, $E_{fish(i,j)}$, may be different for different predators, but in this study the same $E_{fish(i,j)}$ was used for all predators; including black drum, red drum and blue crab. However, $E_{fish(i,j)}$ is different for deposit feeders and suspension feeders, because of the different vertical distribution of these two groups within the soft-bottom habitat.

In addition to standing stock, a second characteristic of prey is its distribution. The feeding rate of predators is expected to increase exponentially when the prey are aggregated in time or space. A "S"-shaped curve is used to simulate this effect (Montagna et al., 1993):

$$E_{fish(i,j)} = 1 - e^{-p_{(5)} \cdot B_{(i,j)}} \tag{15}$$

Where $B_{(i,j)}$ is the biomass of the prey benthos (i=1 or 2 for deposit feeders or suspension feeders, and j=1-8 for eight bays), and $p_{(5)}$ is a new parameter for the aggregation effect. When biomass ($B_{(i,j)}$) is at a very low level, the value of term $E_{fish(i,j)}$ is close to 0, and limitation due to the aggregation effect is nil. When the predator reaches its maximal grazing rate and $B_{(i,j)}$ is very high, the term $E_{fish(i,j)}$ is close to 1, and the limitation due to aggregation is at the maximal level..

2.5. Modeling Tool

The model was constructed using the FORTRAN 77 language and facilitated by the PC software package SENECA (Simulation ENvironment for ECological Application) (de Hoop et al., 1989). SENECA is PC/DOS software package produced by the Netherlands Institute of Ecology. It simplifies model setup, supports techniques for calibration of the model (i.e., estimating the best fit parameter values according to a goodness of fit test), and links programs to a FORTRAN Compiler. The FORTRAN programs created by SENECA are listed in the Appendix.

2.6. Sensitivity of Model

The initial set up of parameter ranges for the model have to be large enough to cover all possibilities. This can be tested with a sensitivity analysis of the initial parameters. The simulation output at the extreme ranges demonstrates that all possible observations of benthic biomass are encompassed by the potential simulations. Therefore, the initial parameter ranges are sensitive enough to include all expected biomass values. (Figs. 2.6.A-Figs. 2.6.D).

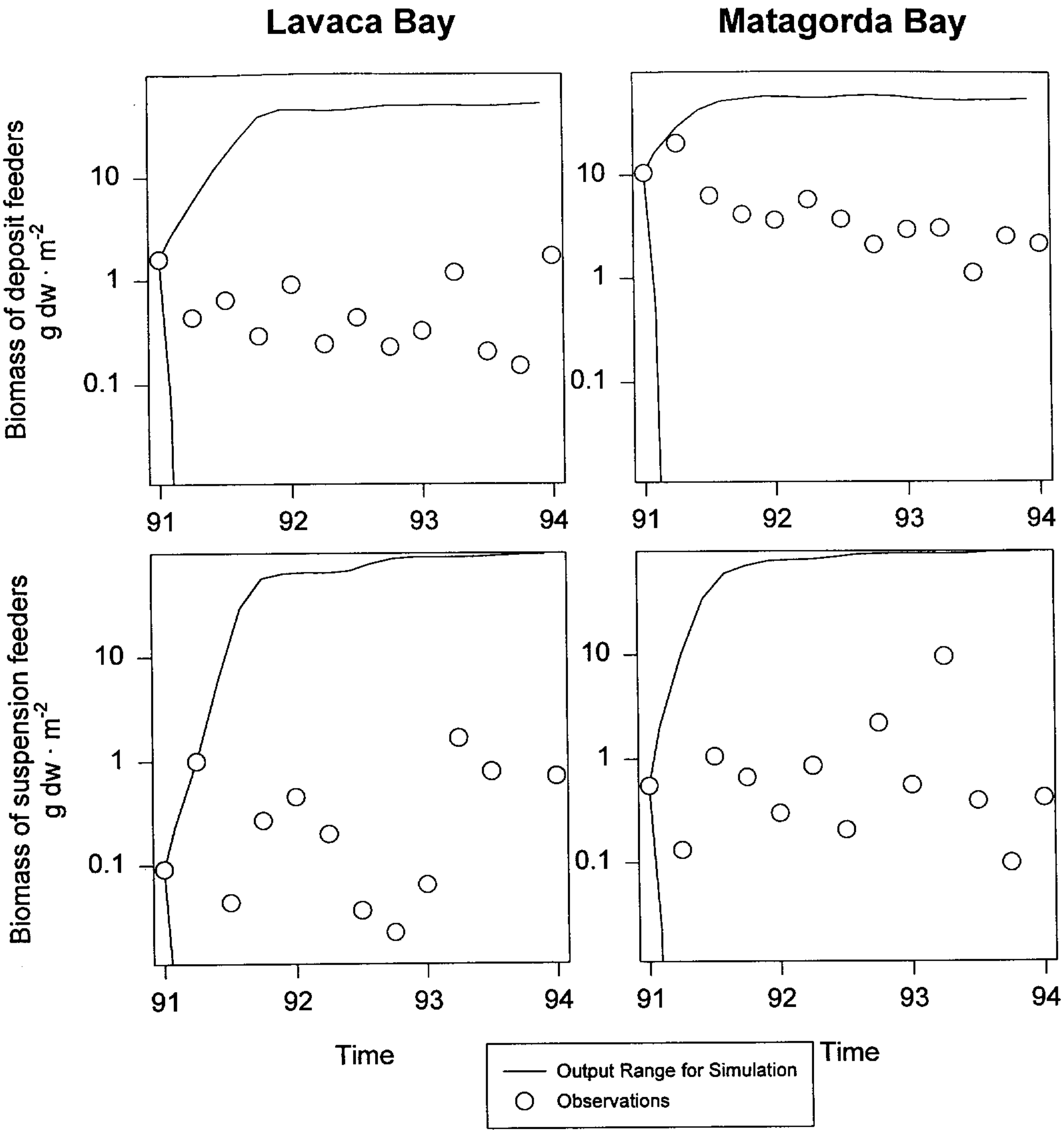


Fig. 2.6.A. Sensitivity analysis for the Lavaca-Colorado Estuary. The output ranges for simulations based 160 sensitivity runs for the 16 initial parameters ranges.

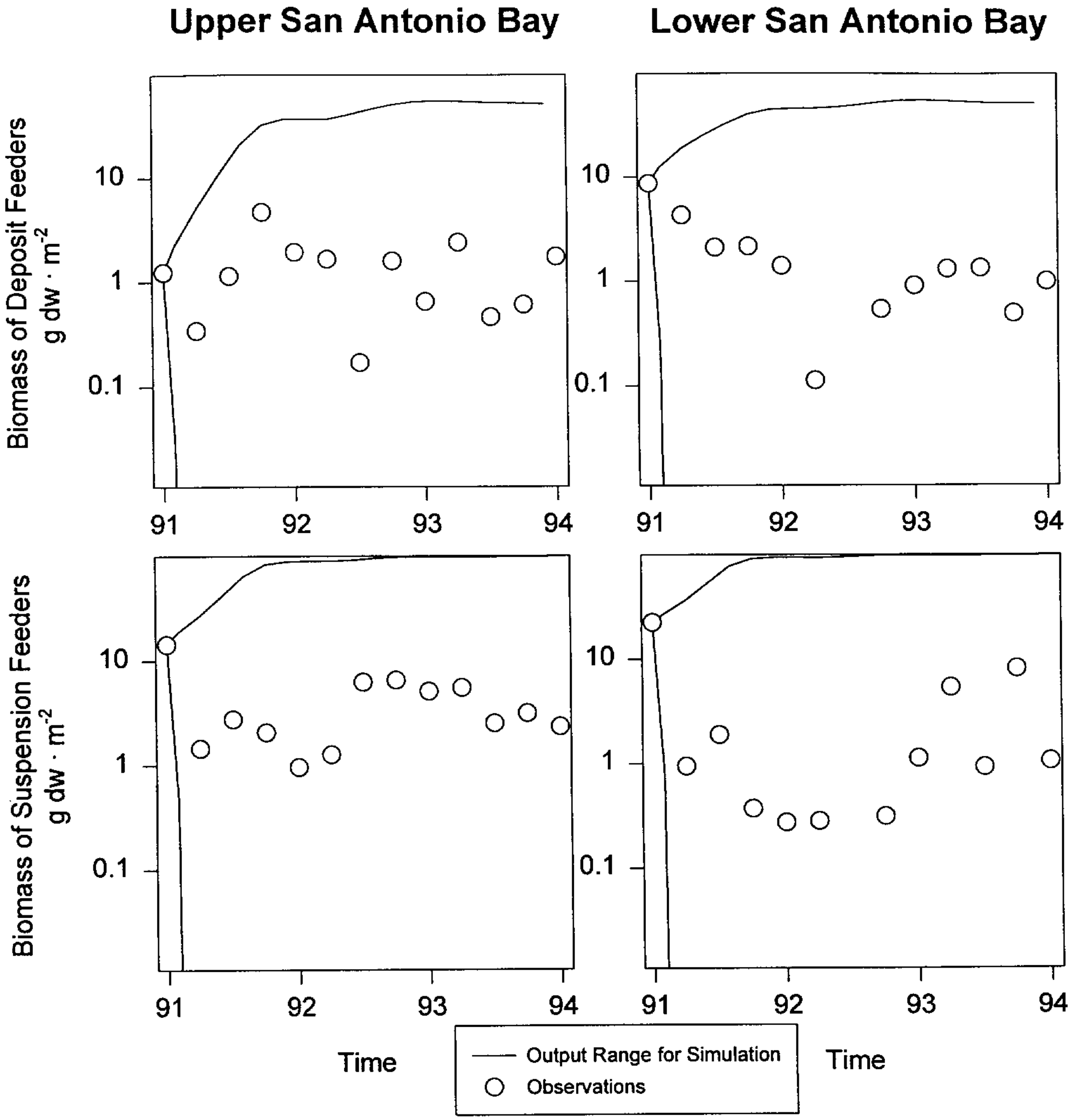


Fig. 2.6.B. Sensitivity analysis for the Guadalupe Estuary. The output ranges for simulations based 160 sensitivity runs for the 16 initial parameters ranges.

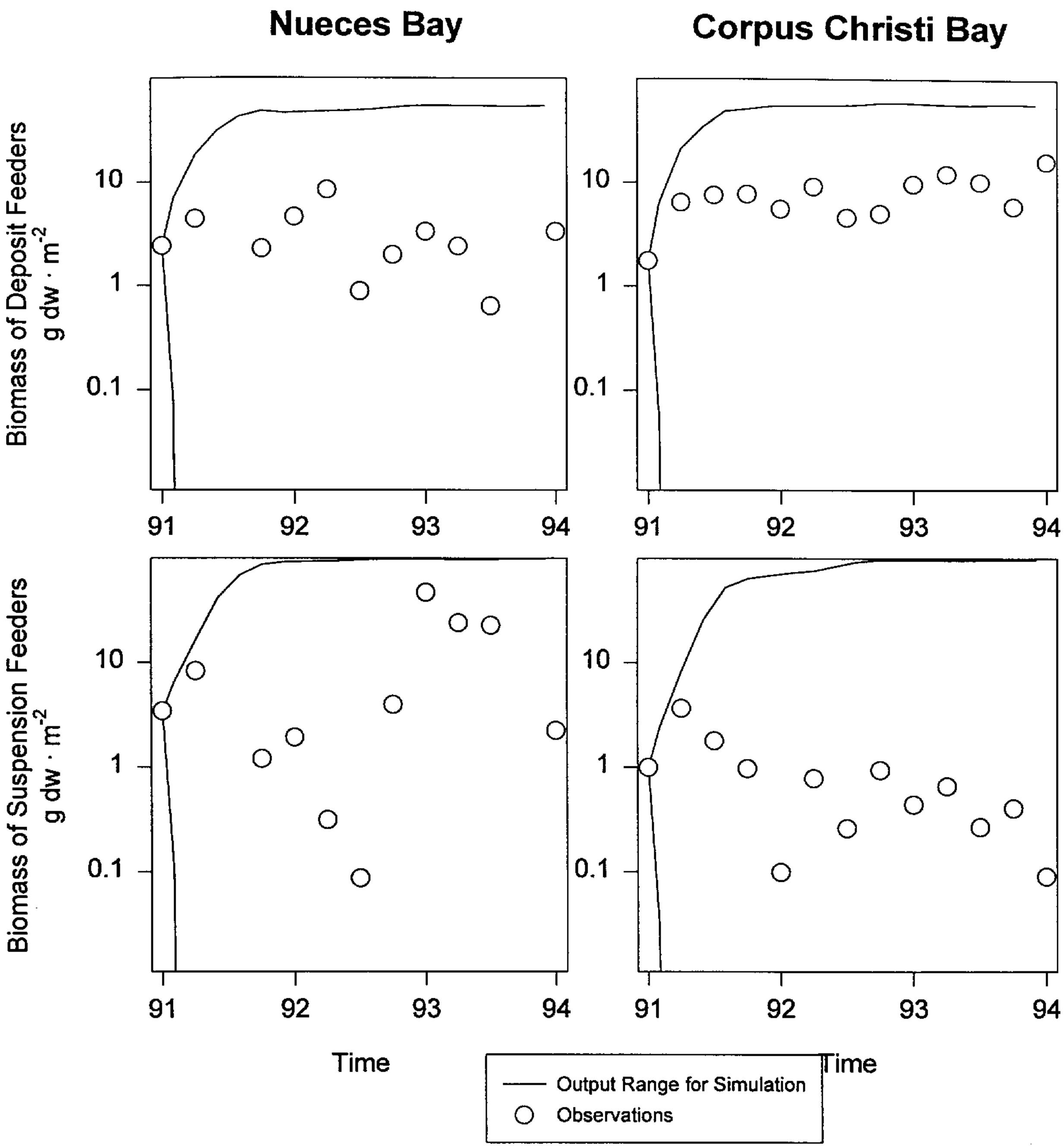


Fig. 2.6.C. Sensitivity analysis for the Nueces Estuary. The output ranges for simulations based 160 sensitivity runs for the 16 initial parameters ranges.

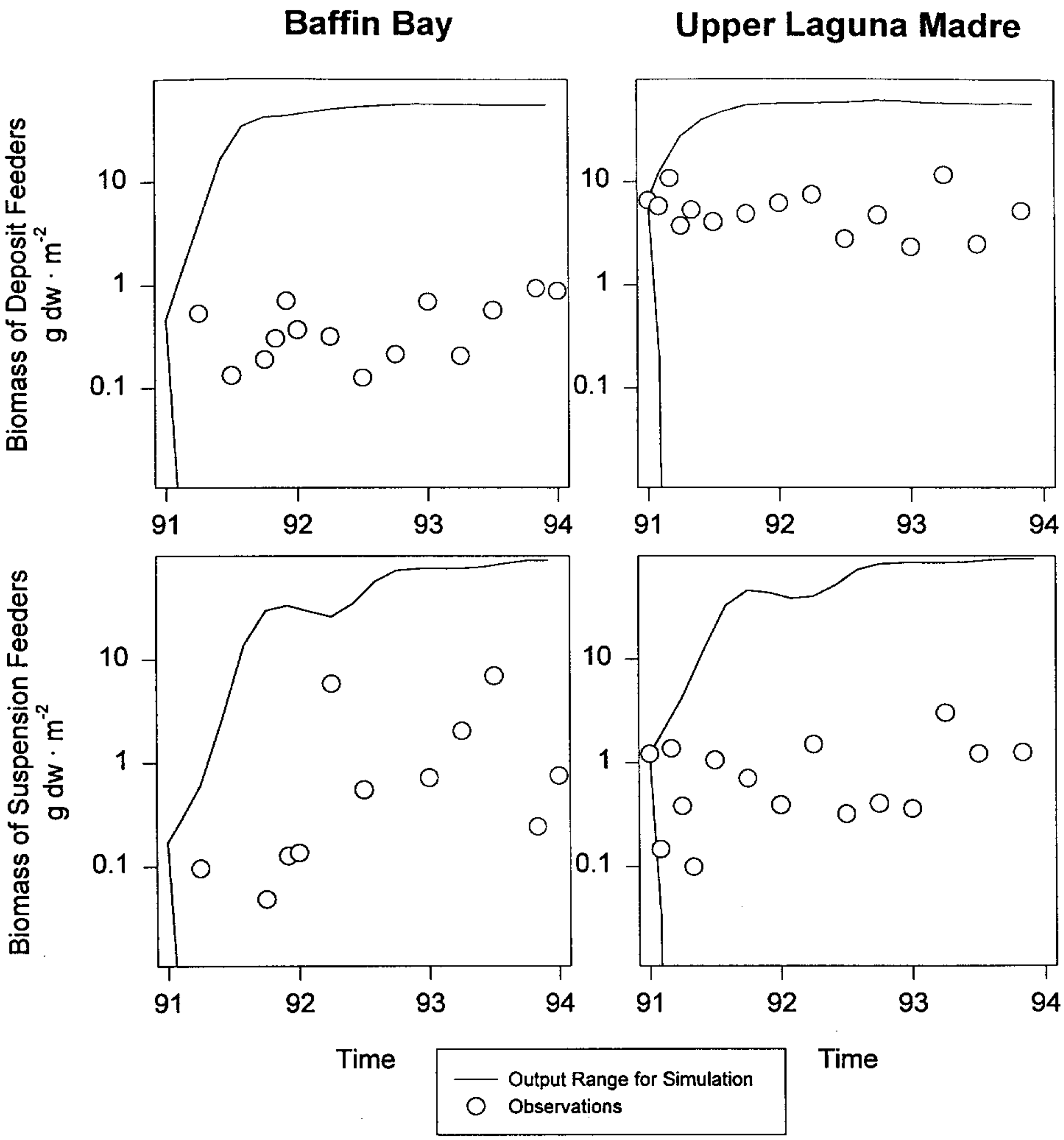


Fig. 2.6.D. Sensitivity analysis for the Laguna Madre Estuary. The output ranges for simulations based 160 sensitivity runs for the 16 initial parameters ranges.

2.7. Calibration of Model

Calibration of the model is a two-step process. Pre-calibration is performed first for those parameters that don't directly affect the benthos simulation. Major calibration of the model is performed after the parameters are fixed from the pre-calibration run.

2.7.1. Pre-calibration

Some simulations were not dependent on the benthos data. A single model was set up for day length, primary production and POM level. This model was calibrated for these parameters, and used in the main model.

2.7.1.1. Day length

Table 2.5 presents the result of calibration for Equation (14) that simulates the day length (Fig. 2.7). The day length simulation is very close to the observed data (Fig. 2.7). These results are used as the parameters in Equation (14) and became a forcing function for the main model.

Table 2.5. Calibration parameters for Equation (14) for the simulation of day length.

Parameter	Definition	Best Fit Value	Reduced Ranges		Initial Ranges	
			Minimum	Maximum	Minimum	Maximum
p_{avg}	Average value of the harmonic function	12.15849	12.15849	12.1585	11	13
p_{amp}	Amplitude of sinus function	1.755811	1.755809	1.755815	1	2
p_{pha}	Phase of sinus function at reference time	0.2244535	0.2244535	0.2244536	0	1

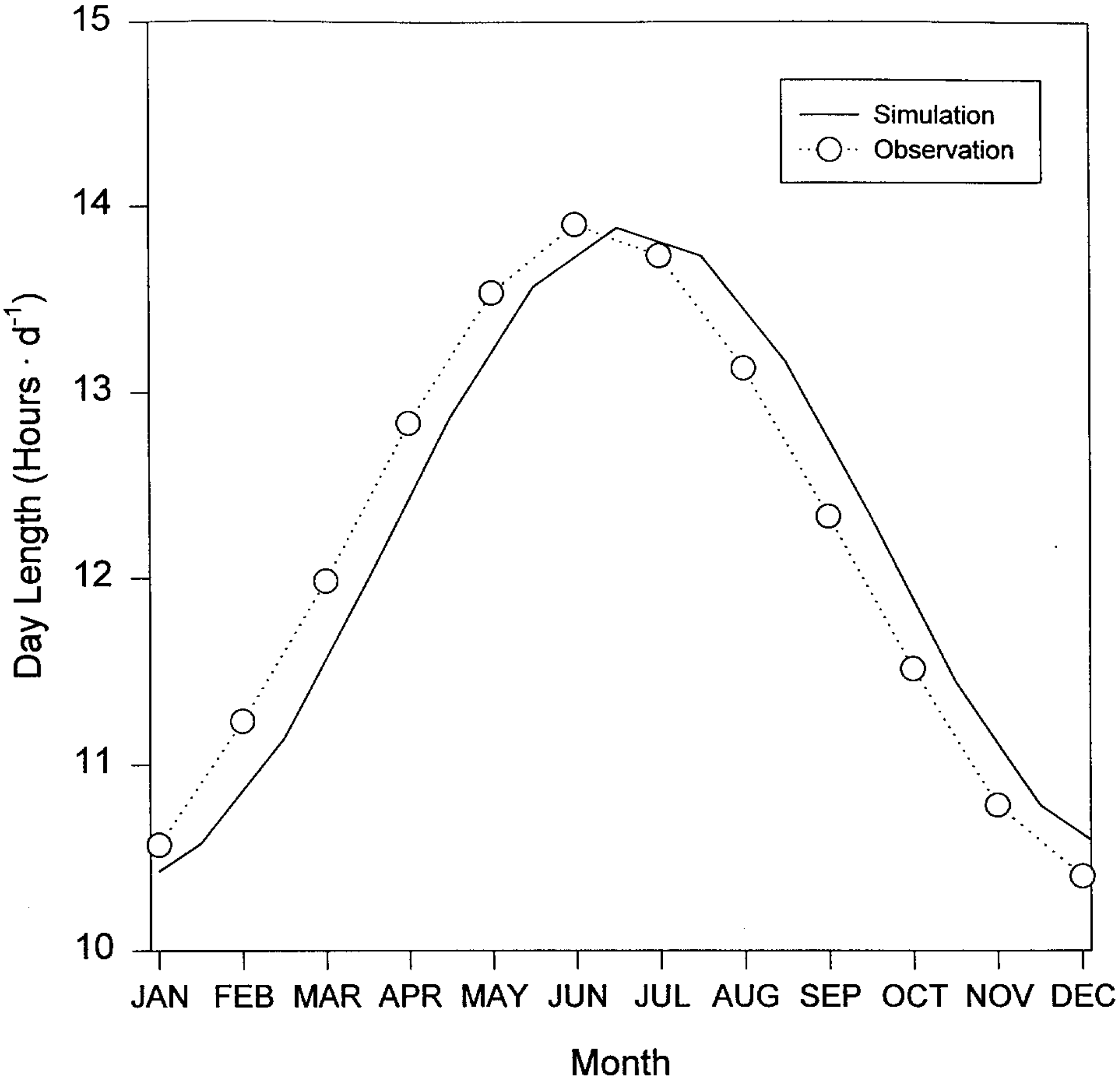


Fig. 2.7. Simulation of day length over a year.

2.7.1.2. Primary Production

Table 2.6. presents the results of the calibration of the parameters for the simulation of primary production that is formulated by equations (12-13). The best fit parameters produce simulations of primary production that corresponds with the range of observations found in other studies (Fig. 2.8.A. and Fig. 2.8.B). Primary production is used as an input for suspension feeders in the main model.

Table 2.6. Calibration of parameters for primary production.

Parameter	Definition	Best Fit Value	Calibrated Ranges		Initial Ranges	
			Minimum	Maximum	Minimum	Maximum
$p_{mic(1)}$	Nutrient limitation	1.909248	1.856677	1.914145	0.5	2
$p_{mic(2)}$	Maximal primary production $(g\ C \bullet m^{-2} \bullet h^{-1})$	0.5093191	0.5076211	0.5105351	0.5	2
$p_{mic(3)}$	Temperature Limitation	11.39727	11.29732	11.39727	10	50

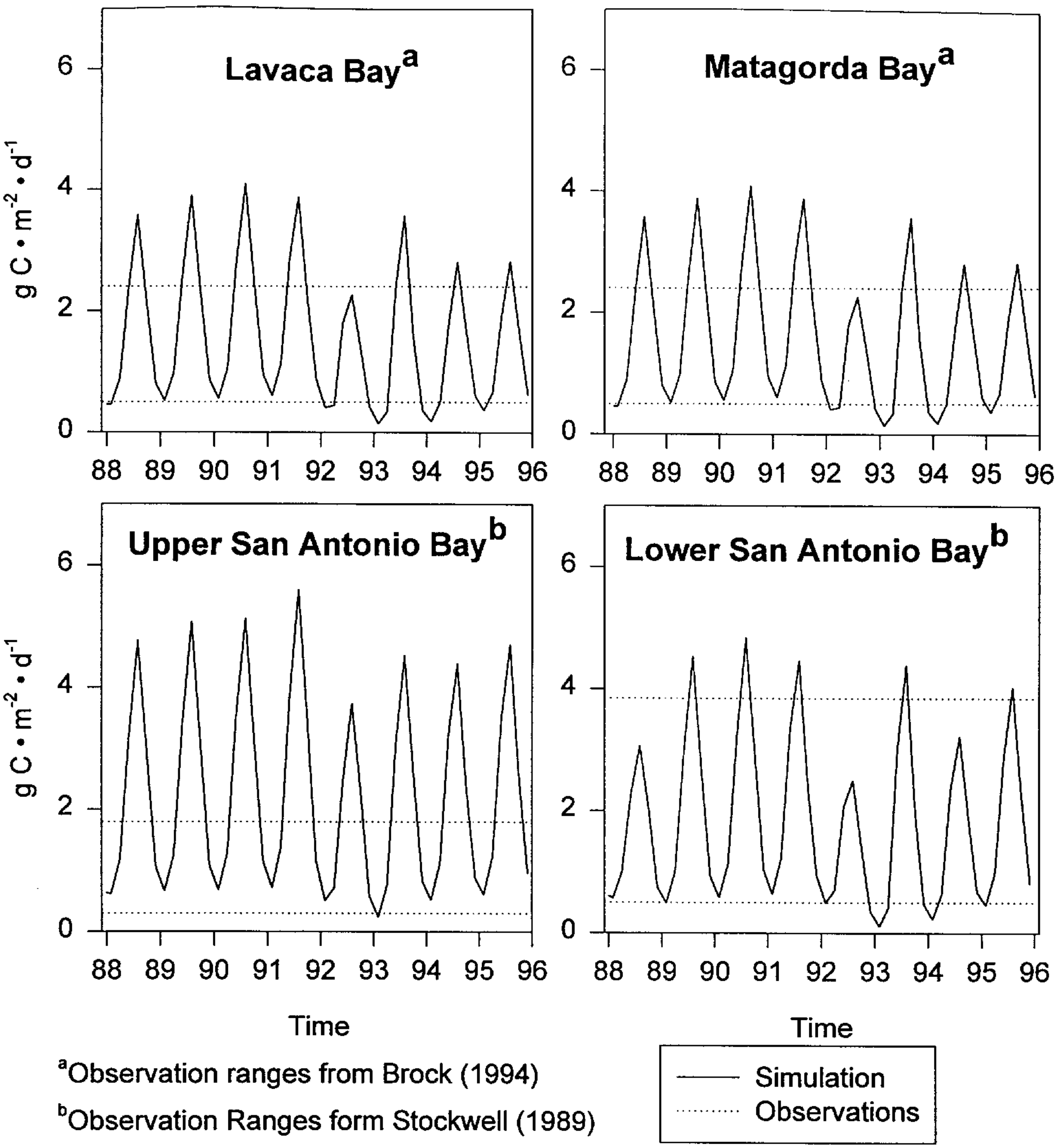
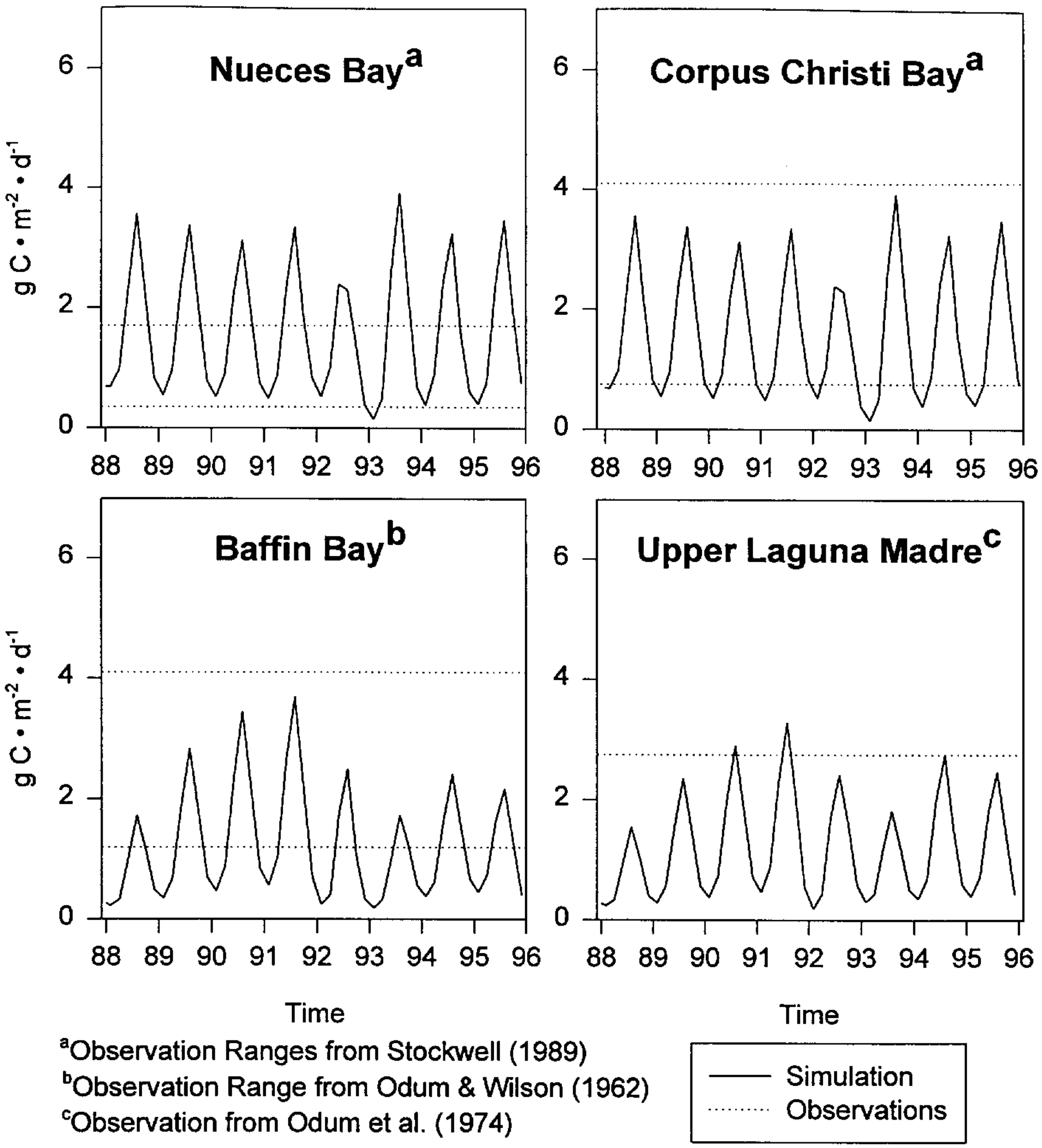


Fig. 2.8.A. Simulation of primary production for Lavaca-Colorado and Guadalupe Estuaries. The simulation is based on the best fit parameters from the model calibration.



[a]Observation Ranges from Stockwell (1989)
[b]Observation Range from Odum & Wilson (1962)
[c]Observation from Odum et al. (1974)

Fig. 2.8.B. Simulation of primary production for Nueces and Upper Laguna Madre Estuaries. The simulation is based on the best fit parameters from the model calibration.

2.7.1.3. POM Level

To calibrate the POM level, we fixed p_{sed} in equation (10) at 18 (Li, et at., 1996), and simulated the carbon concentration. The results of the calibration are listed in Table 2.7, which estimates the carbon concentration levels that were fit to the observations (Figs. 2.9.A-B).

Table 2.7. Parameters for the calibration of POM levels (g dw • m^{-2} •10 cm^{-1}) for the eight bays.

Parameters	Definition	Best Fit Value	Calibrated Ranges		Initial Ranges	
			Minimum	Maximum	Minimum	Maximum
p_{sed}	Sediment Weight (g dw • m^{-2} •10 cm^{-1})	18[a]				
$p_{pom(1)}$	POM level for Lavaca Bay	10136.14	9973.851	10279.69	1000	30000
$p_{pom(2)}$	POM level for Matagorda Bay	10007.6	9589.807	10364.56	1000	30000
$p_{pom(3)}$	POM level for Upper San Antonio Bay	28759.48	28680.92	28819.15	1000	30000
$p_{pom(4)}$	POM level for Lower San Antonio Bay	14611.29	14456.22	14807.55	1000	30000
$p_{pom(5)}$	POM level for Nueces Bay	11070.76	10887.86	11230.25	1000	30000
$p_{pom(6)}$	POM level for Corpus Christi Bay	16988.78	16935.79	17047.56	1000	30000
$p_{pom(7)}$	POM level for Baffin Bay	13804.75	13523.02	14060.89	1000	30000
$p_{pom(8)}$	POM level for Laguna Madre	21193.3	21061.46	21338.11	1000	30000

[a]Calibrated by Li et al.(1996)

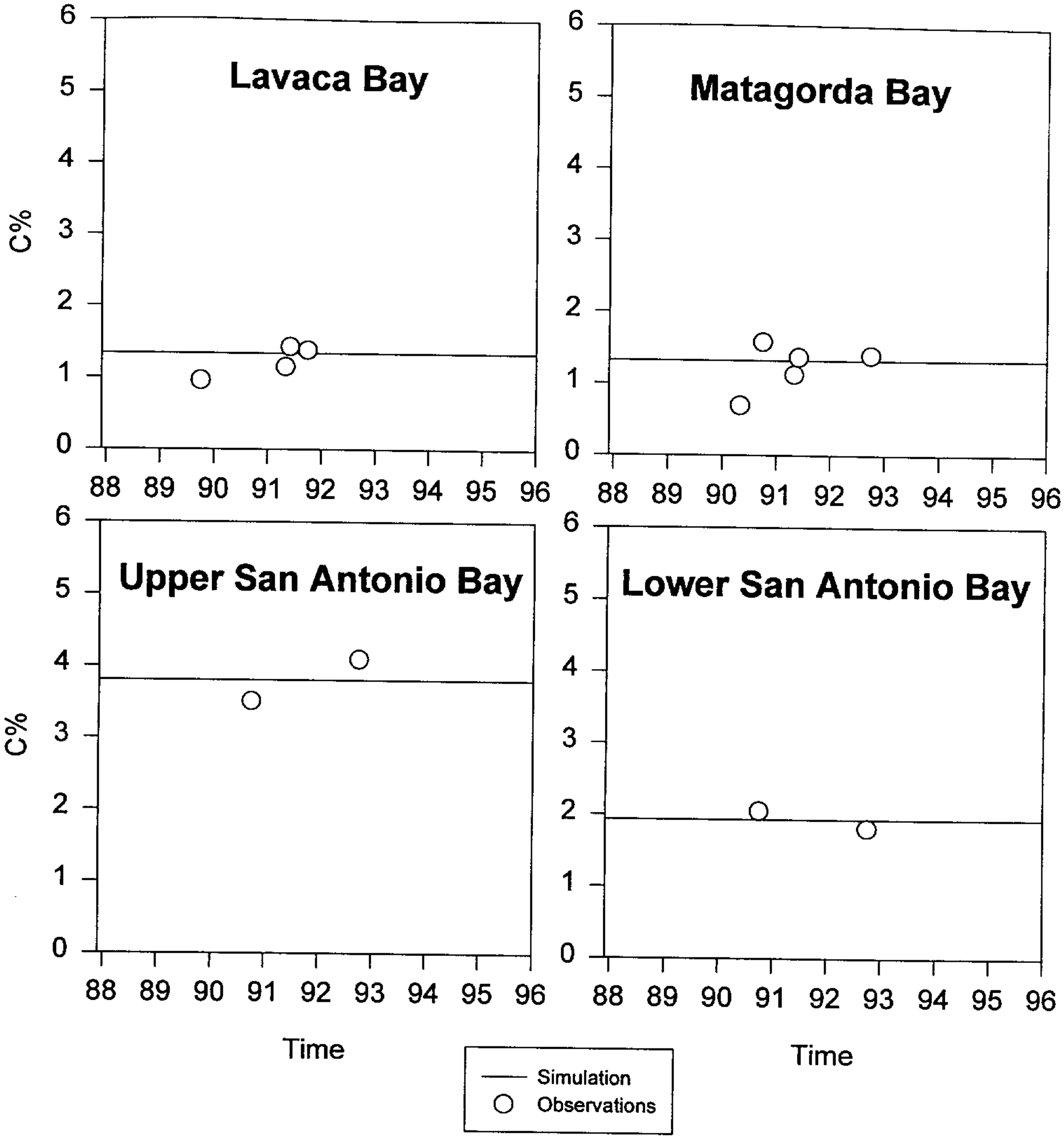


Fig. 2.9.A. The simulation and observations of organic carbon concentration levels in the Lavaca-Colorado and Guadalupe Estuaries.

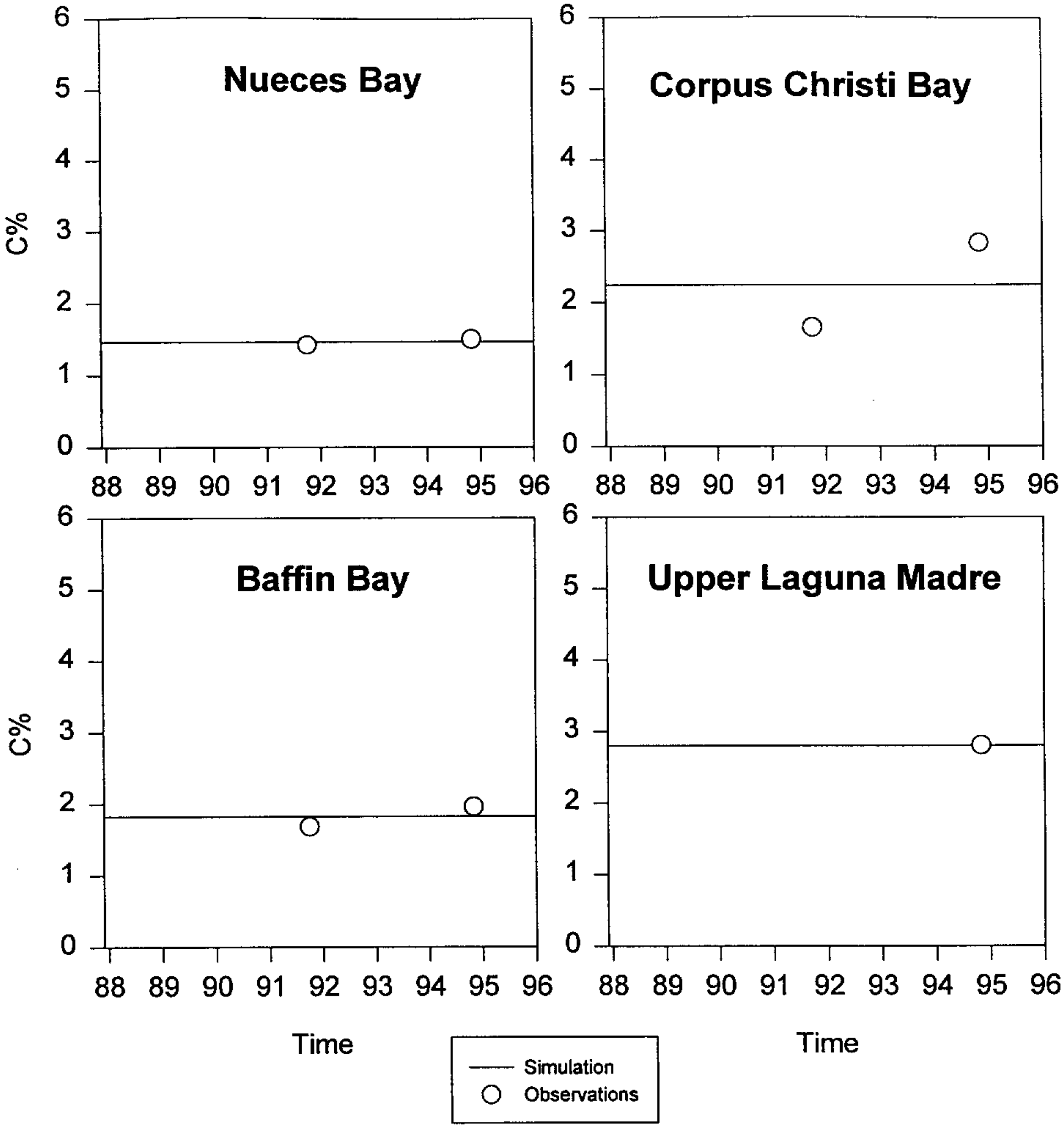


Fig. 2.9.B. The simulation and observations of organic carbon concentrations in the Nueces and Laguna Madre Estuaries.

2.7.2. Major Calibration

The initial calibration of the model was based on the hypothesis that each estuary is one ecosystem consisting of two bays, a primary and secondary bay. This assumes that the environmental limitation parameters are the same for the primary and secondary bays. However, the results of calibration always generated a simulation that was a a poor fit to the observed data. This was especially true of the Guadalupe Estuary. The poor simulation results may be due to a lack of detail in our model, because all estuarine processes are not modeled. Another possibility is that the benthos composition is different in the primary and secondary bay of each estuary. We defined all Polychaeta, Hemicordata, Ophiuroidea and Sipunculida as deposit-feeders, and all Mollusca, Crustacea and Chironomid larvae as suspension-feeders. If the dominant mollusk was a gastropod deposit feeder in one bay and bivalve suspension feeder in another bay, this would yield poor results. Another problem is that the dominant species within a feeding group is different for different bays (Montagna and Kalke, 1992; 1995). For example, the dominant deposit-feeders in Upper San Antonio Bay are *Streblospio benedicti*, *Mediomastus californiensis* and *Hobsonia florida*, but in Lower San Antonio Bay, *Hobsonia florida* is not a dominant species. In Corpus Christi Bay, *Polydora caulleryi* and *Mediomastus californiensis* are dominant deposit-feeders, but *Streblospio benedicti* and *Mediomastus californiensis* dominate in Nueces Bay. The dominant suspension-feeders in Upper and Lower San Antonio Bays are the same: *Littoridina sphinctostoma* and *Mulinia lateralis*. However, the dominant suspension-feeders in Nueces Bay are *Mulinia lateralis* and *Macoma mitchelli*, but they are *Aligena texasiana* and *Leucon sp.* Dominate in Corpus Christi Bay. Ultimately, the model was calibrated for each of the eight bays independently.

There is different data availability for all the bays. The data sets have different periods and different initial levels of the state variables. The four-year period, 1991-1995, is the best data series, with four continuous observations per year in all bays. The longer, seven year period, 1988-1995, has some differences among the eight bays in when the first observation began. Unfortunately, there is a period from 1989 to 1990 when observations for nutrients is missing. However, the longer time series database includes more information of benthos biomass variation as a function of salinity variation. Therefore, the model calibration was separated into two different time periods: the four year series and the seven year series. A comparison between the two different periods of study can be used for an analysis of model validation.

2.7.2.1. Calibration of 1991-1995 Data Series

The model was calibrated eight times for eight bay systems using the period from January 1991 to December 1995. The initial ranges for the seventeen parameters were set within the same values for each bays. There were over 10,000 calibration runs, and all parameter ranges were reduced to less than 50% of the initial ranges. The results of the calibration is presented in the Table 2.8.

2.7.2.2. Calibration of 1988-1995 Data Series

The model was calibrated for the period from April 1988 to December 1995 for Lavaca and Matagorda Bays, from January 1987 to December 1995 for Upper and Lower San Antonio Bays, from December 1987 to April 1995 for Nueces and Corpus Christi Bays, and from March 1989 to December 1994 for Baffin Bay and Laguna Madre. The initial ranges for the seventeen parameters were set within the same values for each bay. There were over 10,000 calibration runs and all parameters ranges were reduced to less than 50% of the initial ranges. The result of the calibration is presented in the Table 2.8.

Table 2.8. Best fit parameter values from the calibration of eight bays systems for the continuous four year data base: 1991-1995. Abbreviations: 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. LB did not have red drum and black drum observations, so the parameters $g_{(i,1,1)}$ and $g_{(i,1,2)}$ were not computed. The parameters are defined in Equations (4-15).

Para-meter	Best fit values for each Estuary								Initial ranges
	LB	MB	US	LS	NB	CC	BB	LM	
$p_{(1)}$	44.732329	42.42344	44.57255	41.4347	40.24104	44.2046	42.54965	42.3392	20, 45
$p_{(2)}$	19.66252	17.13879	17.77497	39.31757	17.91409	17.7302	32.6675	35.26918	20, 45
$p_{(1,3)}$	34.88885	37.76986	27.05453	34.99224	29.95359	29.62446	35.79842	32.04066	20, 40
$p_{(1,4)}$	96.10882	62.42303	2.550676	44.03107	74.82873	4.951183	93.46018	85.88029	0, 100
$g_{(1,j,1)}$	-	0.5528514	3.944138	3.096442	3.175888	0.6201606	0.1099941	4.909842	0, 5
$g_{(1,j,2)}$	-	3.470691	4.975888	4.823431	4.743777	4.972863	2.613795	4.958814	0, 5
$g_{(1,j,3)}$	0.6814387	1.247655	0.1387833	0.07430105	0.4850444	1.664876	1.450189	0.7158198	0, 5
$p_{(5)}$	2.05926E-3	3.7239E-3	1.32259E-3	9.22492E-3	4.17089E-3	2.85868E-3	2.28463E-3	2.08676E-3	0.001, 0.01
$r_{(2)}$	5.80609	6.670129	5.074876	5.077178	6.316035	6.259595	5.101094	6.540565	5, 20
$c_{(1)}$	68.88308	30.30947	69.80885	51.28811	42.26977	30.6242	46.33809	34.76485	30, 70
$p_{(2,3)}$	16.74664	5.10743	8.106342	19.25299	7.11083	9.867107	17.58461	14.17485	20, 40
$p_{(2,4)}$	6.457159	15.1268	89.89465	40.01582	81.45561	44.78239	85.16875	64.63729	0, 100
$g_{(2,j,1)}$	-	3.705585	0.8315539	4.858335	3.312729	1.334923	0.0976517	3.820603	0, 5
$g_{(2,j,2)}$	-	0.4145367	0.5318332	4.515656	4.939147	0.400967	0.635004	4.975064	0, 5
$g_{(2,j,3)}$	0.9929386	1.195466	0.4320632	0.09369156	0.4154133	1.625607	1.46157	0.5934232	0, 5
$r_{(2)}$	5.343904	8.033044	5.942176	5.186666	7.632288	7.890732	6.403854	6.627981	5, 20
$c_{(2)}$	50.24743	99.65412	89.25981	52.67889	87.83488	97.43611	52.60339	76.37729	50, 100

Table 2.9. Best fit parameter values from the calibration of eight bays systems for the full seven year data base: 1991-1995. Abbreviations: as in Table 2.8. Lavaca Bay does not have observations of red drum and black drum, so the parameters $g_{(i,1,1)}$ and $g_{(i,1,2)}$ are not calculated The parameters here are defined in Equations (4-15).

Para-meters	Best fit values for each estuary								Initial ranges
	LB	MB	US	LS	NB	CC	BB	LM	
$p_{(1)}$	42.17162	41.71341	40.53684	40.33885	42.30378	40.0247	44.86441	43.71011	20, 45
$p_{(2)}$	17.11376	17.11468	17.10742	35.23304	22.73746	19.141	24.33928	37.73854	20, 45
$p_{(1,3)}$	31.3853	35.47797	29.20075	33.51442	34.95972	27.12825	27.15673	28.57515	20, 40
$p_{(1,4)}$	31.69317	45.49924	66.75111	70.64457	31.90294	99.50521	70.44145	65.51883	0, 100
$g_{(1,j,1)}$	-	1.564318	4.975216	4.977599	1.627888	0.7996559	0.1418857	0.9445796	0, 5
$g_{(1,j,2)}$	-	0.4472847	0.1738868	1.56211	1.921271	0.2069192	4.975177	3.199269	0, 5
$g_{(1,j,3)}$	0.9455477	3.39694	0.07164717	0.2115675	2.221616	2.341787	2.610957	0.5797191	0, 5
$p_{(5)}$	1.69155E-3	2.26646E-3	2.94225E-3	3.44199E-3	1.2195E-3	1.64854E-3	1.36815E-3	3.6306E-3	0.001, 0.01
$r_{(2)}$	8.506821	11.38221	5.336619	5.074531	5.087093	5.088249	5.367772	7.747459	5, 20
$c_{(1)}$	43.29486	30.25233	61.2493	38.85413	40.78039	46.40247	62.28746	52.16617	30, 70
$p_{(2,3)}$	5.29994	5.095006	5.073665	5.449434	19.61624	9.719198	19.92501	10.07507	20, 40
$p_{(2,4)}$	68.00172	34.75121	36.48948	79.79934	88.95782	31.06932	7.293406	92.97546	0, 100
$g_{(2,j,1)}$	-	0.09629202	3.648983	0.7421896	4.389579	0.9922819	0.2115792	2.727237	0, 5
$g_{(2,j,2)}$	-	2.034983	4.975049	4.628137	3.487815	4.278396	4.223227	1.683713	0, 5
$g_{(2,j,3)}$	1.637504	2.048566	0.1243953	0.2623813	4.3093	2.76253	4.96872	0.5738177	0, 5
$r_{(2)}$	5.074632	7.75984	5.074776	5.224783	10.46929	8.323019	9.536337	11.29916	5, 20
$c_{(2)}$	99.75256	60.42086	96.94084	99.75688	82.12787	71.37692	51.4902	63.78191	50, 100

2.7. Model Validation

Model validation can be accomplished by three methods: goodness of fit test values, biological meaning of the estimated parameters, and comparison of the different simulation periods for the same model structure.

The period from 1991 to 1995 is a period in which synoptic data was measured in all eight bays. This data period is the best for comparing the eight bays. There are two differences between the short-term synoptic simulation and long-term simulation: the number of observations are different, and the initial time and values for the state variables are different (Table 2.10). Using the best fit parameters calibrated from the short-term synoptic database (1991-1995) for the simulation of the long-term period (1987-1995) is a method for validation of the model. In reverse, the best fit parameters calibrated by long-term database in each bay system can also be used to simulate the synoptic short-term measurements of benthic biomass.

Table 2.10. Initial values (mg dw · m^{-2}) for the state variables for the two simulation periods. The periods are: 1991-1995 and 1988-1995, and the observations are for deposit-feeders and suspension-feeders. Abbreviations as in Table 2.8.

Bays	Synoptic Short-term Period (1991-1995)			Long-term Period (1988-1995)		
	Initial Date	Deposit-feeders	Suspension-feeders	Initial Date	Deposit-feeders	Suspension-feeders
LB	January 1991	1.592	0.091	April 1988	1.452	3.462
MB	January 1991	10.484	0.554	April 1988	10.150	2.275
US	January 1991	1.237	14.197	January 1987	1.300	0.960
LS	January 1991	8.652	22.303	January 1987	0.885	0.706
NB	January 1991	2.416	3.447	December 1987	0.324	0.026
CC	January 1991	1.755	1.000	December 1987	1.263	3.465
BB	January 1991	0.461	0.096	March 1989	0.310	2.684
LM	January 1991	6.510	1.210	March 1989	7.332	29.404

2.7.1. Goodness of Fit Test Values

The goodness of fit test compares each simulation against the observed values. The model is valid when the variance of the goodness of fit test is close to zero. The goodness of fit values for all simulations or parameters calibrated by short-term or long-term data sets ranged from 0.59 to 3.25 (Table 2.11).

The simulation of Corpus Christi Bay has the best goodness of fit values, followed by Lower San Antonio Bay, Matagorda Bay, Laguna Madre, Baffin Bay, Nueces Bay, and Upper San Antonio Bay. Lavaca Bay has worst goodness of fit values. The short-term simulation has a better fit than the long-term simulation. The short-term simulation was not better using parameters calibrated from the long-term database than those from the short-term database. Surprisingly, the long-term simulation was better using the parameters calibrated from the short-term database than parameters calibrated using the long-term database. This was true for both benthic feeding types in Lavaca Bay (2.39<3.25 and 1.45<1.63) and Lower San Antonio Bay (0.84<0.89 and 0.95<1.05), but only true for suspension-feeders in Matagorda Bay (1.00<1.08) and deposit-feeders in Corpus Christi Bay (0.84<0.92). In general, the short-term synoptic data simulation is more appropriate for use in this study rather than the long-term data simulation.

Table 2.11. The goodness of fit values for simulation and model validation. Simulation is performed using the parameters calibrated from same period and validation is performed using parameters calibrated from different periods. The short-term synoptic period was 1991-1995, and the long-term period (which is different for each bay) is 1988-1995. Test values are given for both feeding types: deposit and suspension feeders. Abbreviations as in Table 2.8.

Bay	Simulation				Validation				Average of all tests
	Short-term		Long-term		Short-term		Long-term		
	Deposit	Suspension	Deposit	Suspension	Deposit	Suspension	Deposit	Suspension	
LB	1.94	1.06	3.25	1.63	3.25	2.04	2.39	1.45	2.28
MB	0.96	0.81	1.04	1.08	0.93	1.39	1.21	1.00	1.23
US	0.91	0.75	0.94	1.01	1.79	2.28	1.15	1.23	1.61
LS	0.75	1.07	0.89	1.05	1.06	1.34	0.84	0.95	1.05
NB	0.72	0.59	0.91	1.13	1.03	2.76	1.26	1.14	1.55
CC	0.63	0.63	0.92	0.87	0.96	1.29	0.84	0.92	1.00
BB	0.81	0.70	0.73	1.15	1.03	1.19	1.18	1.62	1.26
LM	0.80	0.75	1.47	0.79	1.26	1.08	1.49	1.14	1.24

2.7.2. Biological Responses

The suspension-feeders in all eight bays have smaller suitable salinity ranges than deposit-feeders in each bay (Tables 2.8 and 2.9). Therefore, the model should predict higher production to biomass (P/B) ratios for suspension feeders in secondary bays than in primary bays. The predicted annual P/B ratios are similar for deposit-feeders in all eight bays. In the case of suspension-feeders, the annual P/B ratios are always higher in secondary bays than in primary bays (Table 3.1). Therefore, the model structure is valid in that it correctly predicts expected biological responses based on know biological mechanisms. An invalid model structure would predict a biological response that is counter intuitive.

2.7.3. Comparison of Simulations With Different Calibration Periods

The model should be able to simulate any time period if the model structure is valid. The simulation of the 1991-1995 period using the calibration results of the data from the 1988-1995 period are stable for Matagorda, Nueces, Corpus Christi, Upper and Lower San Antonio, Baffin Bays and Upper Laguna Madre (Figs. 2.10. A-D). The simulation of the 1988-1995 period using the calibration results of the data from the 1991-1995 period are also stable for Matagorda, Corpus Christi, Upper and Lower San Antonio, Baffin Bays and Upper Laguna Madre (Figs. 2.11 A-D). The worst simulation is for both deposit-feeders and suspension-feeders biomass in the Lavaca Bay.

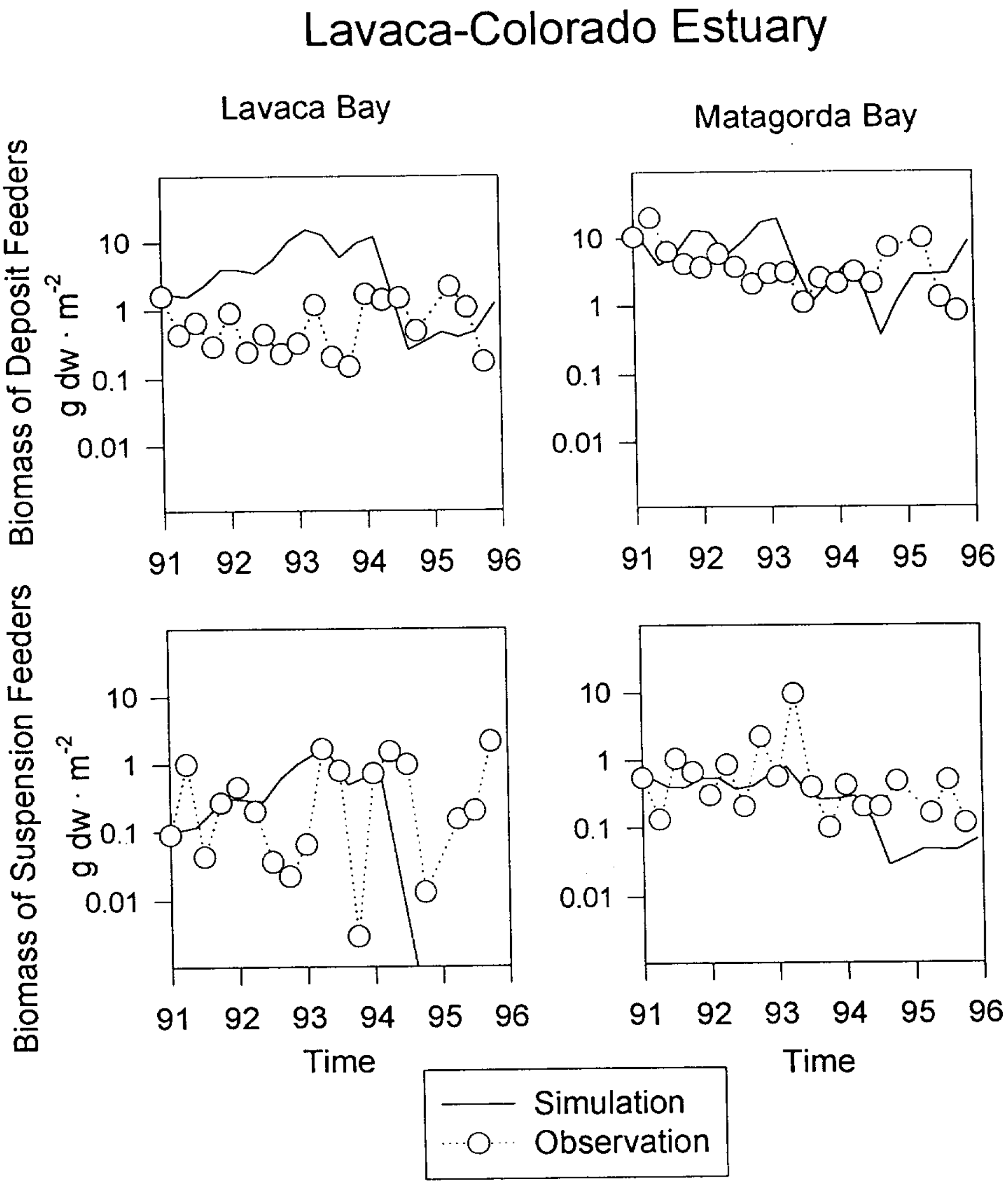


Fig. 2.10.A. The simulation of benthos biomass for the period 1991-1995 using the results of the calibration of the period 1988-1995 for Lavaca-Colorado Estuary.

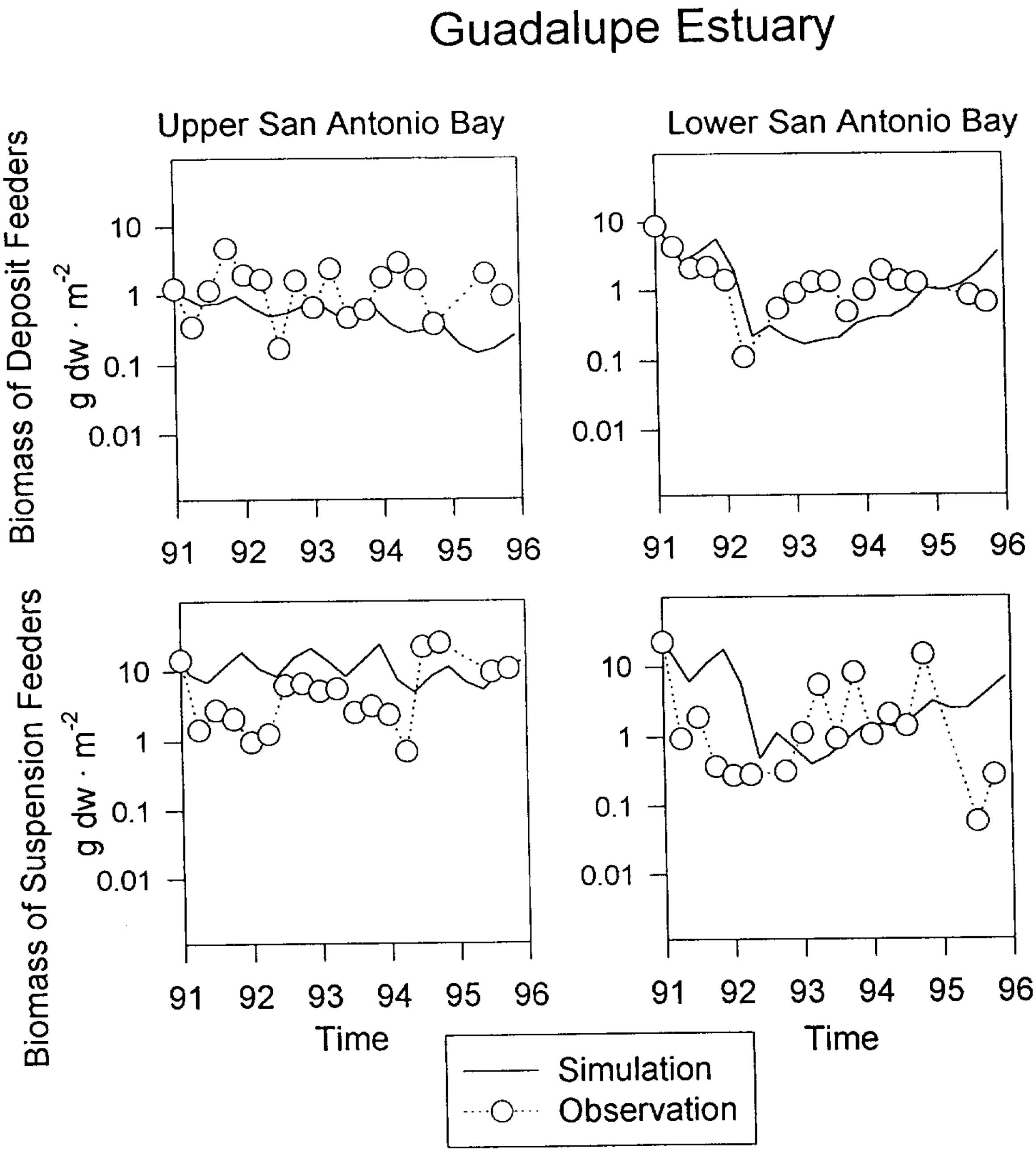


Fig. 2.10.B. The simulation of benthos biomass for the period 1991-1995 using the results of the calibration of the period 1987-1995 for Guadalupe Estuary.

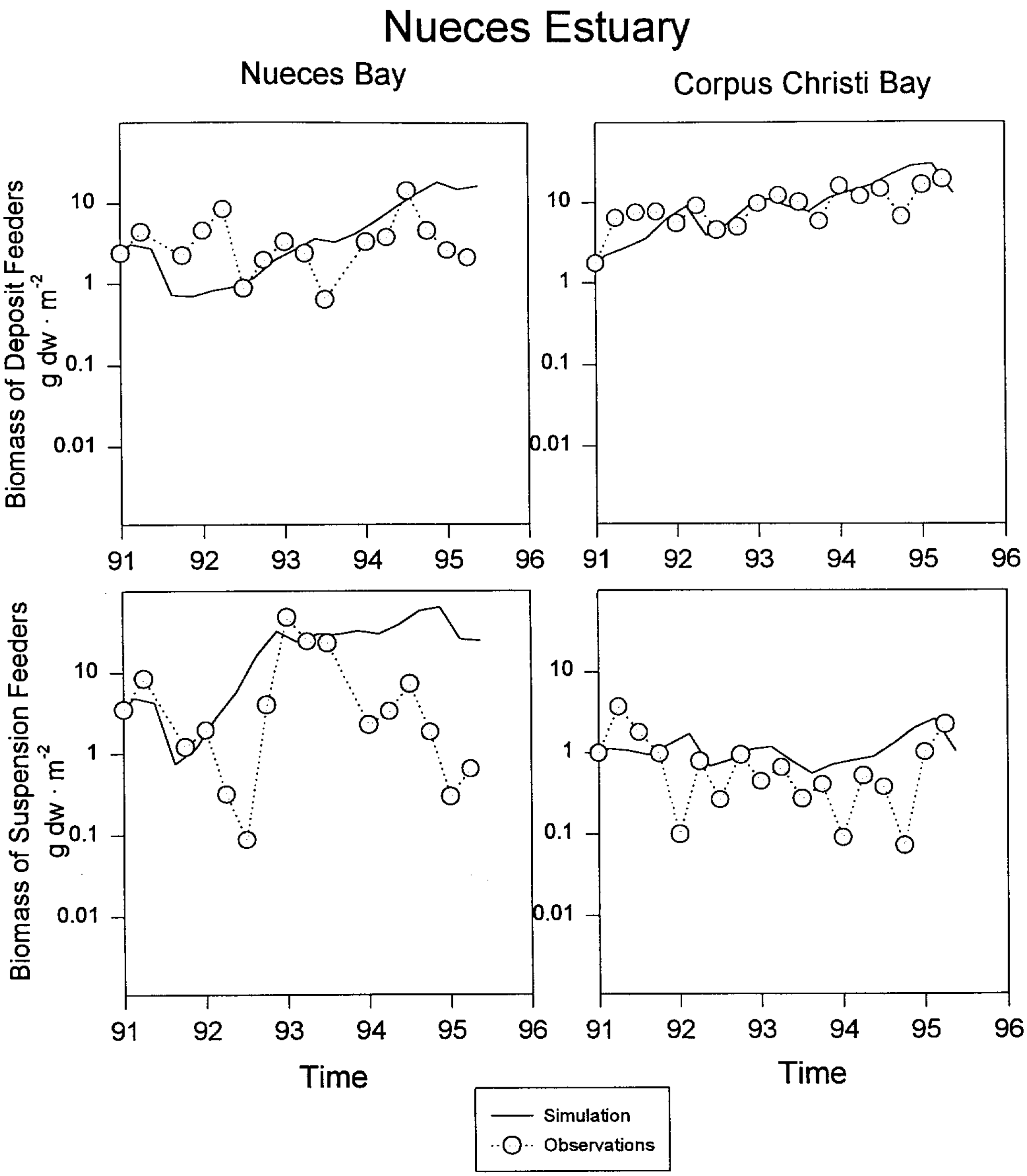


Fig. 2.10.C. The simulation of benthos biomass for the period 1991-1995 using the results of the calibration of the period 1987-1995 for Nueces Estuary.

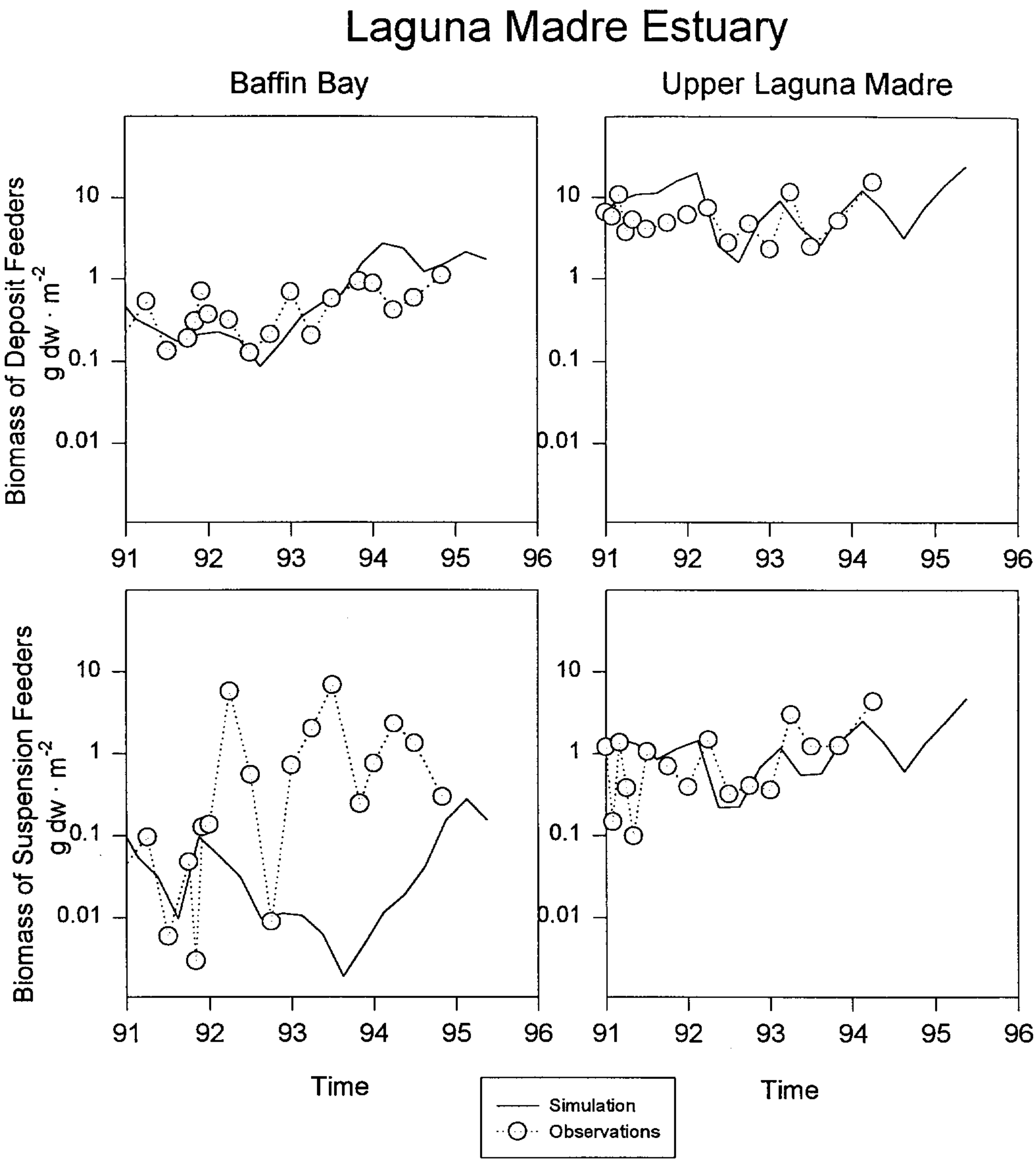


Fig. 2.10.D. The simulation of benthos biomass for the period 1991-1995 using the results of the calibration of the period 1989-1995 for Upper Laguna Madre Estuary.

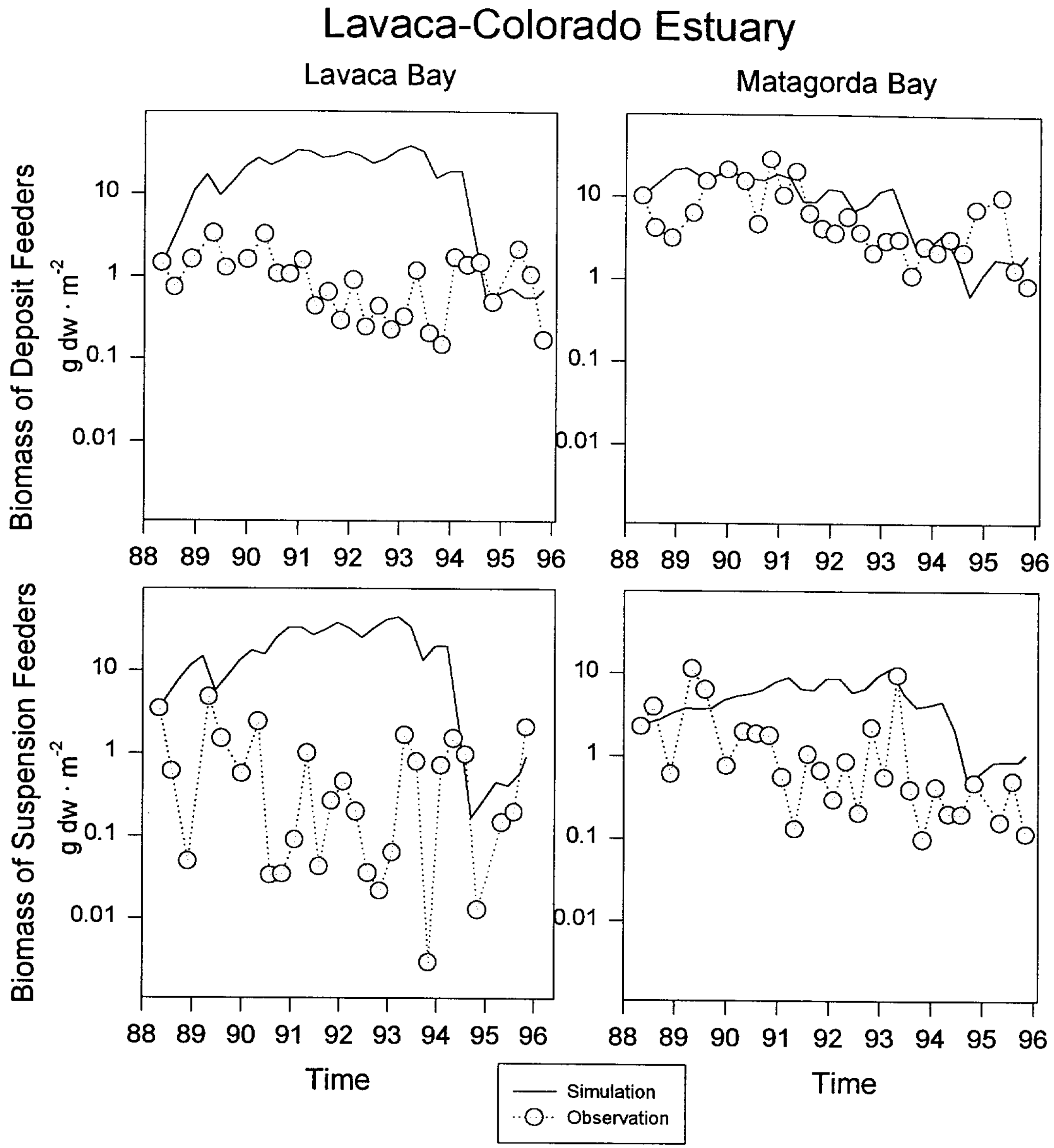


Fig. 2.11.A. The simulation of benthos biomass for the period 1988-1995 using the results of the calibration of the period 1991-1995 for Lavaca-Colorado Estuary.

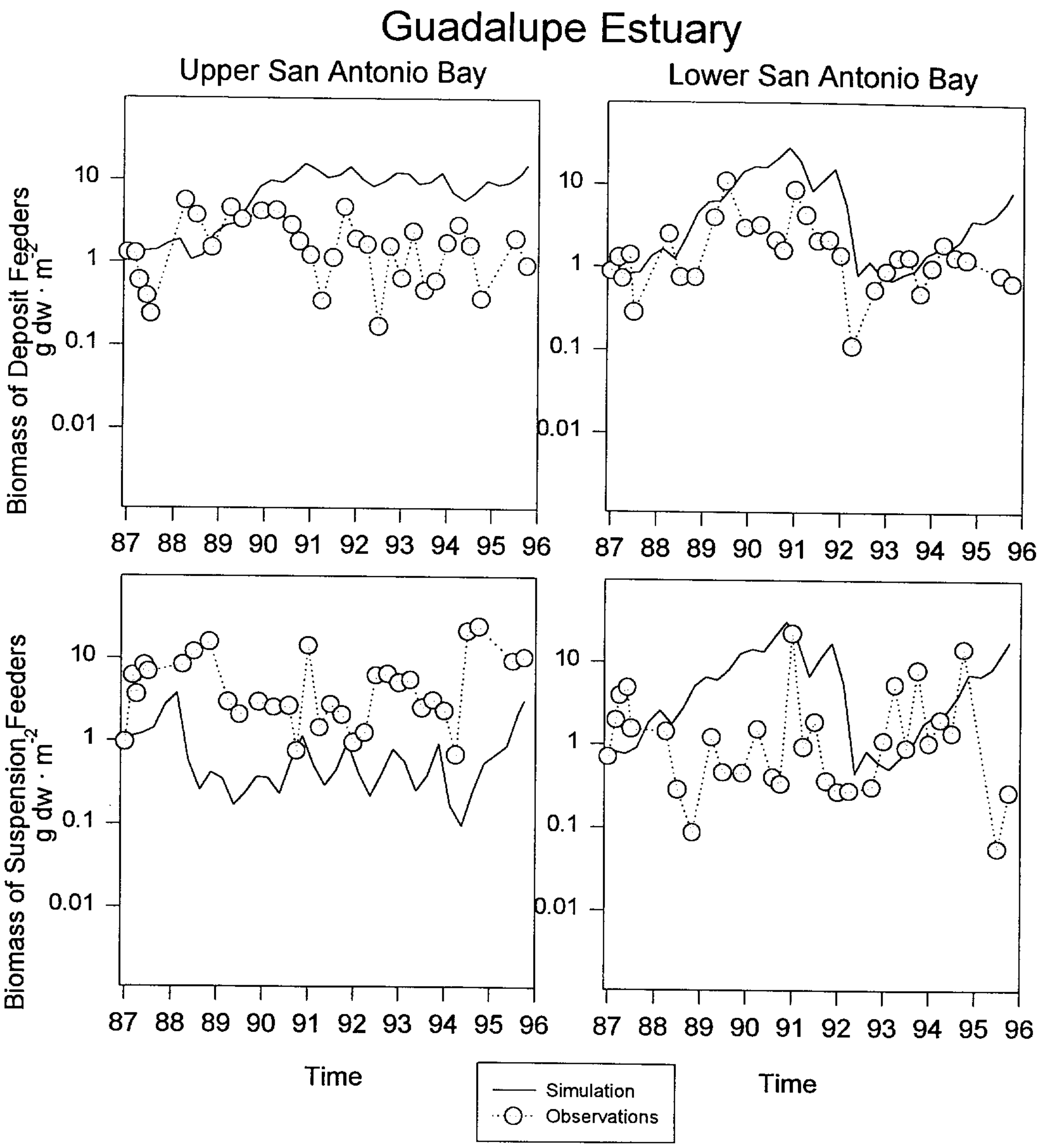


Fig. 2.11.B. The simulation of benthos biomass for the period 1987-1995 using the results of the calibration of the period 1991-1995 for Guadalupe Estuary.

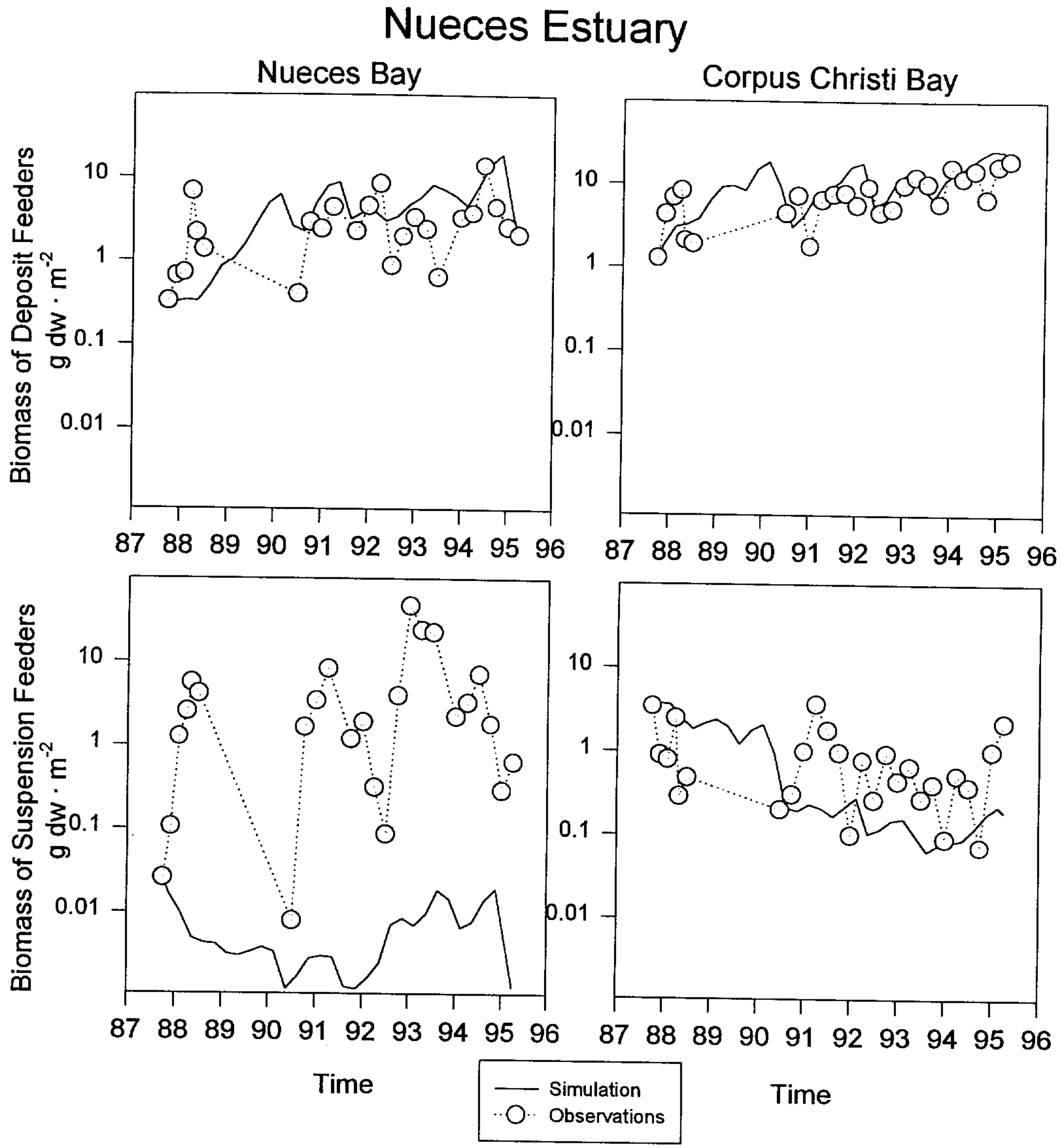


Fig. 2.11.C. The simulation of benthos biomass for the period 1987-1995 using the results of the calibration of the period 1991-1995 for Nueces Estuary.

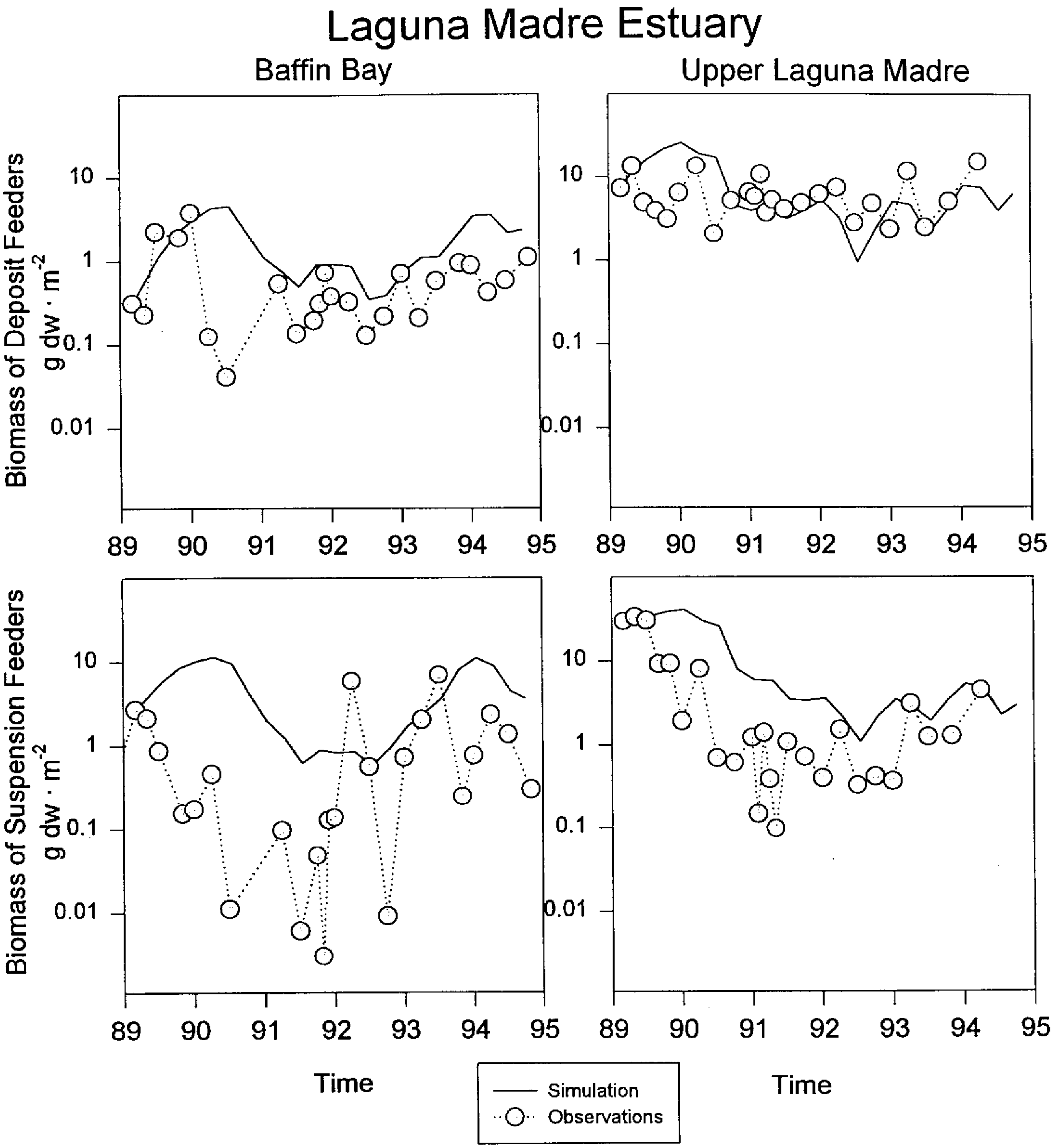


Fig. 2.11.D. The simulation of benthos biomass for the period 1989-1995 using the results of the calibration of the period 1991-1995 for Upper Laguna Madre Estuary.

3. RESULTS

The simulations of benthic biomass are based on the best fit parameters from the calibration of the period (1991-1995). The average values of biomass over the entire simulation period are the expected values of the benthos standing stocks. Production, annual P/B, production efficiency, and environmental limitation were all based on the calibrations and simulations of the period from 1991 to 1995.

3.1. Simulation of Benthos Biomass

The simulations of benthos biomass for deposit-feeders and suspension feeders were partly successful (Figs. 3.1.A-D). The simulations fit well for most bays, except for Lavaca Bay. The values for the goodness of fit tests demonstrate the different levels of performance of the simulations (Table. 2.11). Simulations for both feeding types that had goodness of fit values lower than 1 include: Matagorda Bay, Upper San Antonio Bay, Nueces Bay, Corpus Christi Bay, Baffin Bay and Upper Laguna Madre. In Lower San Antonio Bay, the goodness of fit value for the simulation of deposit-feeders is lower than 1 while it is 1.07 for suspension-feeders. Both simulations of the two feeding types in Lavaca Bay are higher than 1, and even near 2, for deposit-feeders.

Suspension-feeders have higher simulated biomass variation through the 1991-1995 period over all eight bays than do deposit-feeders (Figs. 3.1.A-D). The primary bays, closest to the sea, such as Matagorda, Lower San Antonio, Corpus Christi Bays and Laguna Madre, have more stable biomass for both feeding types than the secondary bays, close to the freshwater sources, such as Lavaca, Lower San Antonio, Nueces and Baffin Bays.

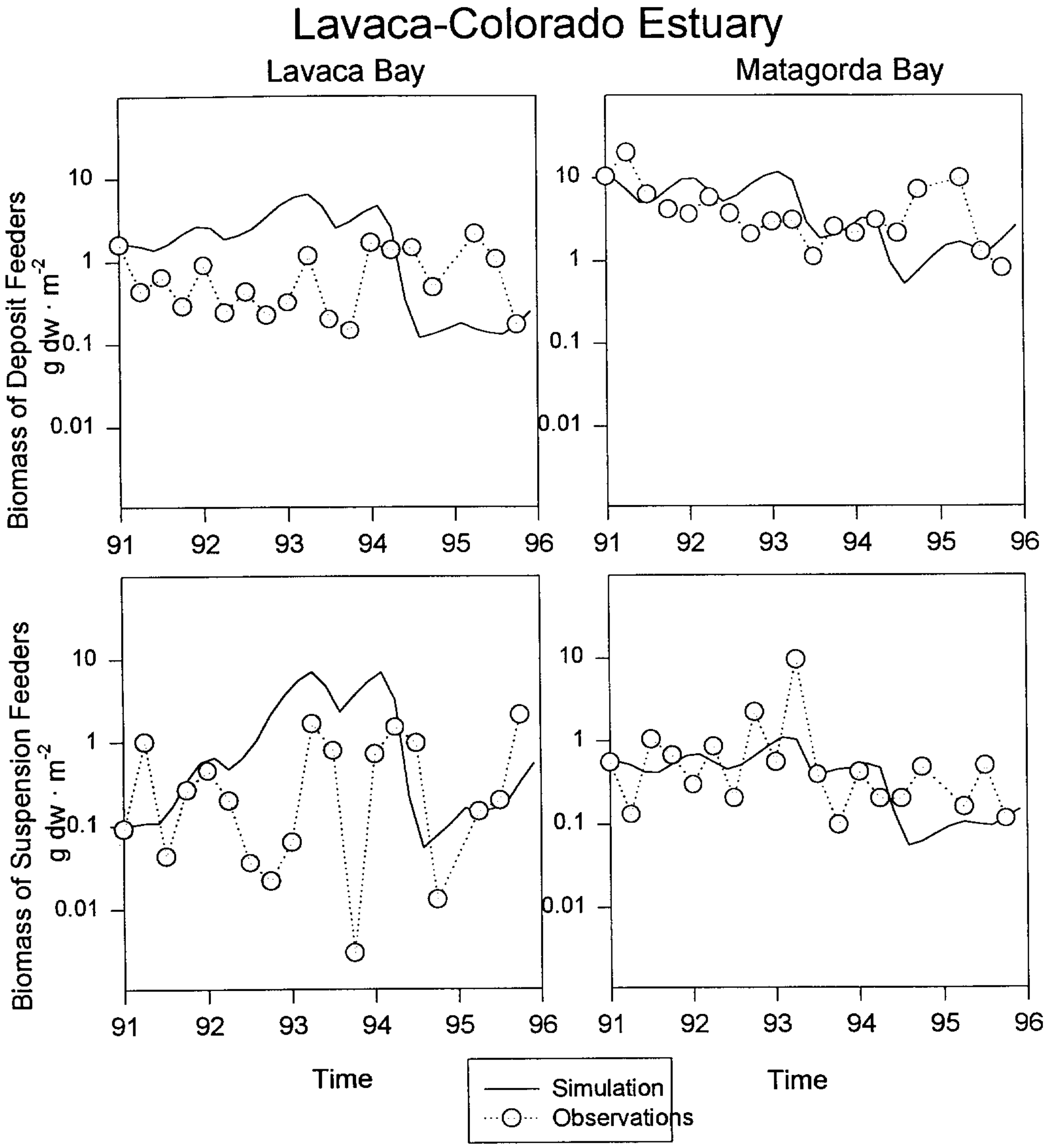


Fig. 3.1.A. Simulation of deposit-feeders and suspension-feeders biomass in the Lavaca-Colorado Estuary for the period 1991-1995.

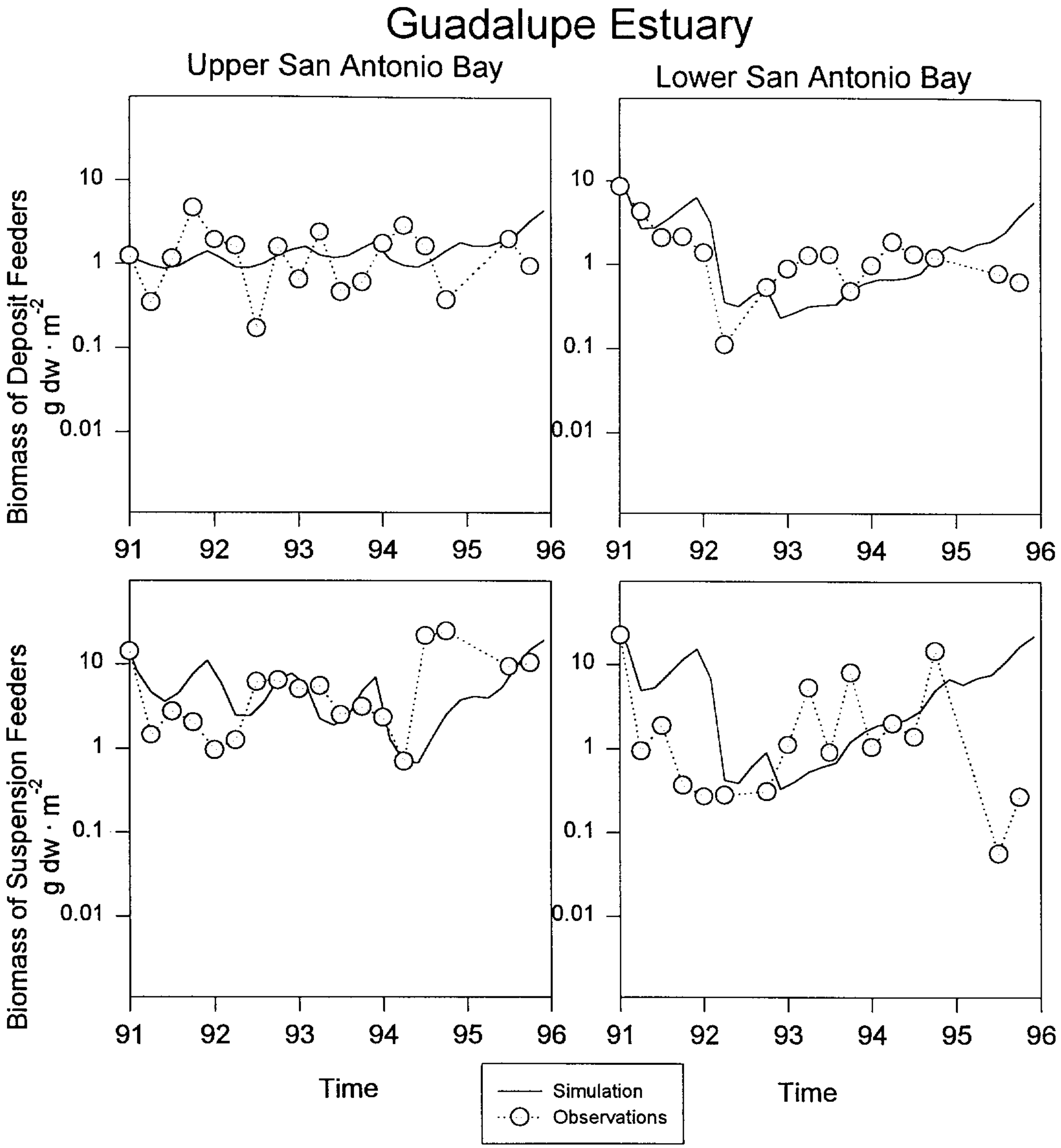


Fig. 3.1.B. Simulation of deposit-feeders and suspension-feeders biomass in the Lavaca-Colorado Estuary for the period 1991-1995.

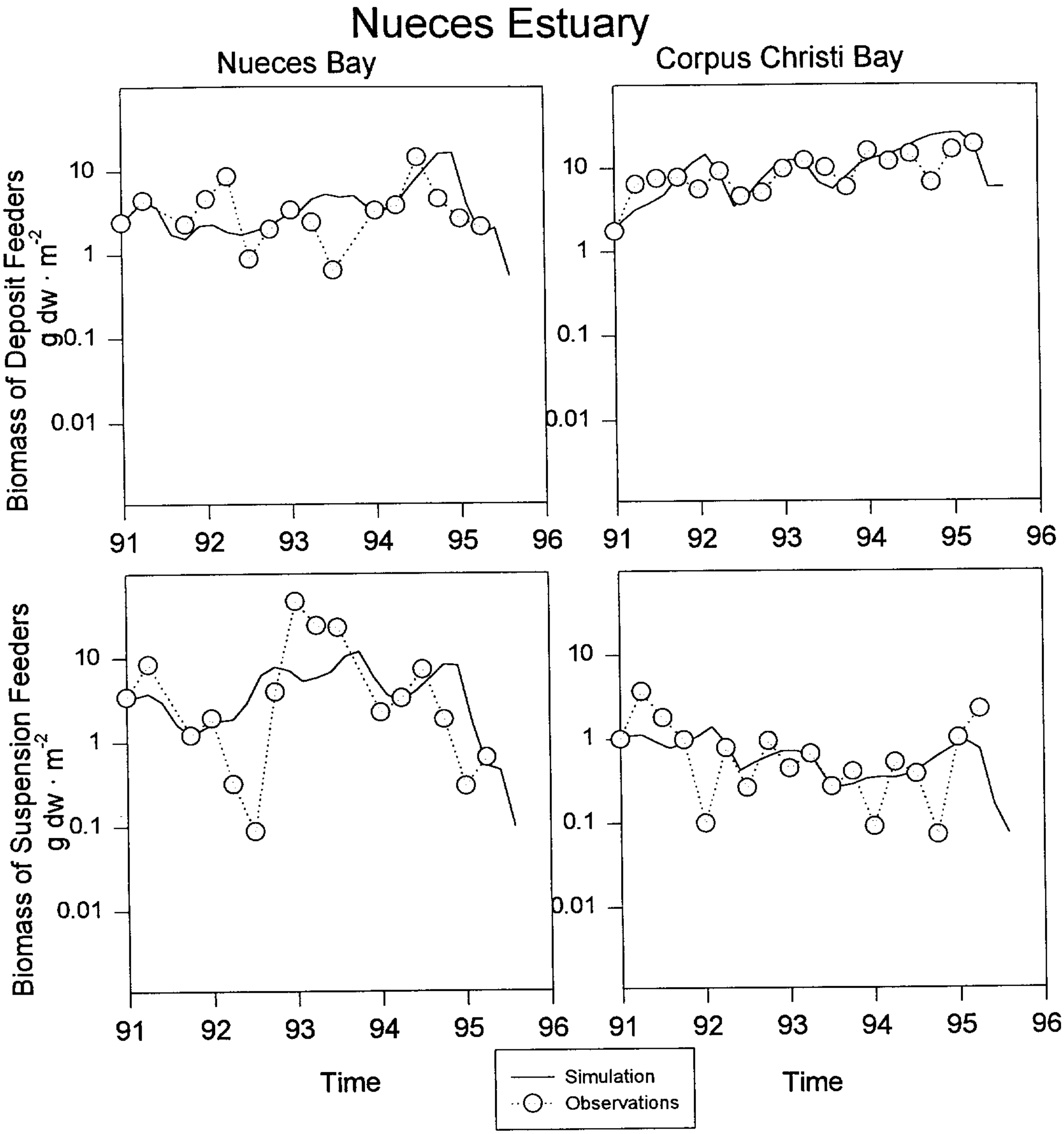


Fig. 3.1.C. Simulation of deposit-feeders and suspension-feeders biomass in the Lavaca-Colorado Estuary for the period 1991-1995.

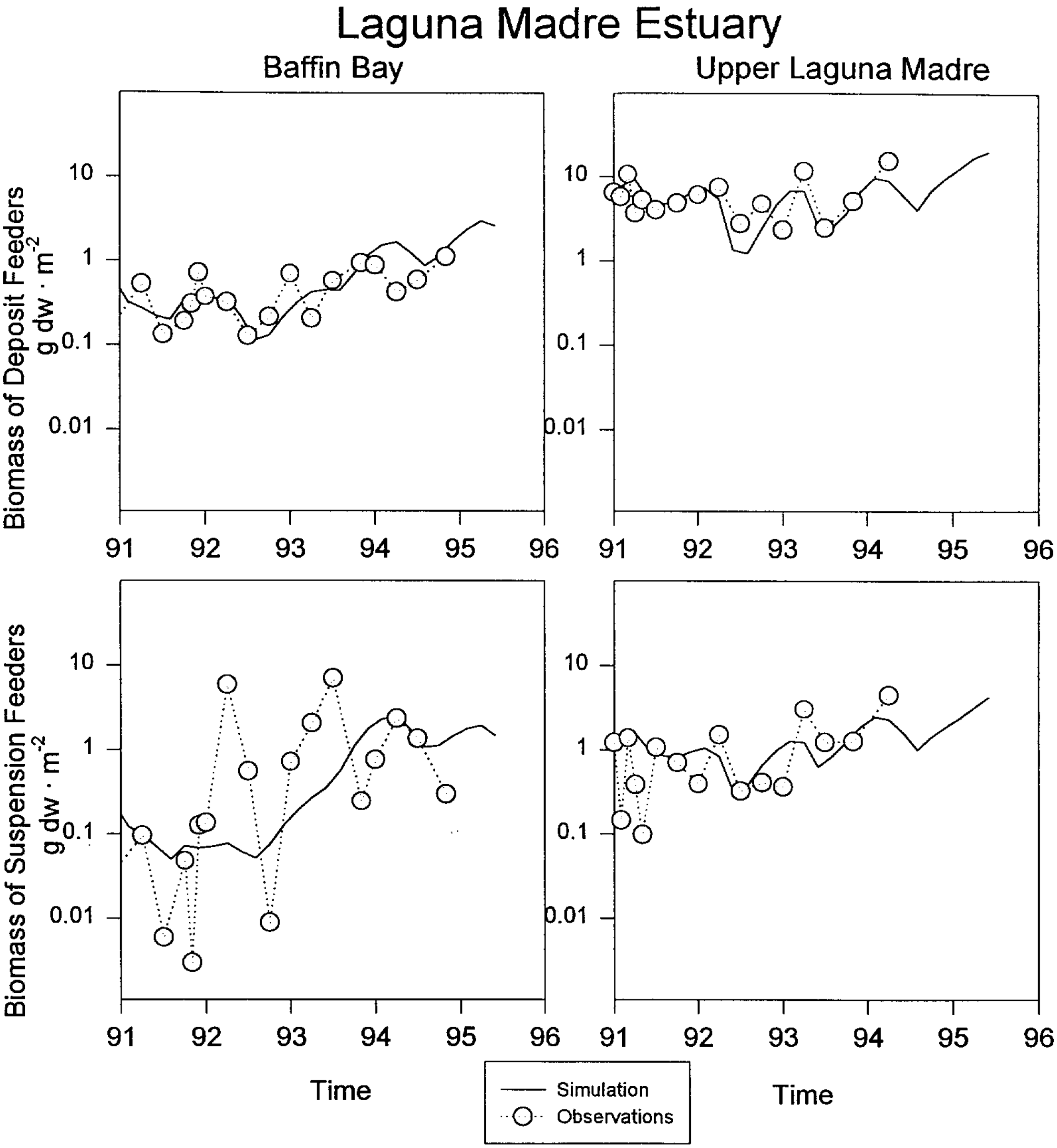


Fig. 3.1.D. Simulation of deposit-feeders and suspension-feeders biomass in the Laguna Madre Estuary for the period 1991-1995.

3.2. Standing Stocks

The average benthic biomass simulated by the model is the predicted standing stock of benthos in the eight bays. The standing stocks in the Nueces Estuary during 1991-1995, which is 9.1-11.8 g dw · m^{-2}, is higher than in other estuaries (Table 3.1). It is followed by Guadalupe Estuary (6.6-7.7 g dw · m^{-2}). Upper Laguna Madre has a median level standing stock at 6.7 g dw · m^{-2}. Lavaca-Colorado Estuary has lower level at 3.9-5.4 g dw · m^{-2} and Baffin Bay has the lowest level in the study area at only 1.2 g dw · m^{-2}.

Overall, it appears that the secondary bays close to the freshwater source have higher levels of suspension feeder standing stocks, while the primary bays near the sea have higher standing stock levels of deposit feeders.

3.3. Secondary Production

The average production rate of benthos in Texas estuaries ranges from 3.4-24.3 g dw · m^{-2} · y^{-1} during the simulation period of 1991-1995 (Table 3.1). Nueces Estuary has the highest production level due to both deposit feeders and suspension feeders, which have annual production rates of over 11 g dw · m^{-2} in Nueces Bay and a much higher level of deposit feeders production in Corpus Christi Bay of 22.9 g dw · m^{-2} · y^{-1}. Guadalupe Estuary has a medium level of production in the range of 16.6-18.2 g dw · m^{-2} · y^{-1} due to a higher level of production by suspension-feeders (13.8-14.4). Laguna Madre Estuary has a large difference of production between Upper Laguna Madre and Baffin Bay. In the Upper Laguna Madre, the production of deposit feeders is as high as 18.3 g dw · m^{-2} · y^{-1}, which is second only to Corpus Christi Bay and 20 times higher than in Baffin Bay.

The long-term temporal variation of monthly production rates during the period 1991-1995 is simulated by the model (Figs. 3.2. A-B). Monthly production is more stable for deposit feeders than for suspension feeders in Nueces Bay, but the reverse is true in the Upper Laguna Madre. The monthly production rate has a large decrease for both feeding types in 1994 for Lavaca Bay and in 1992 for Lower San Antonio Bay.

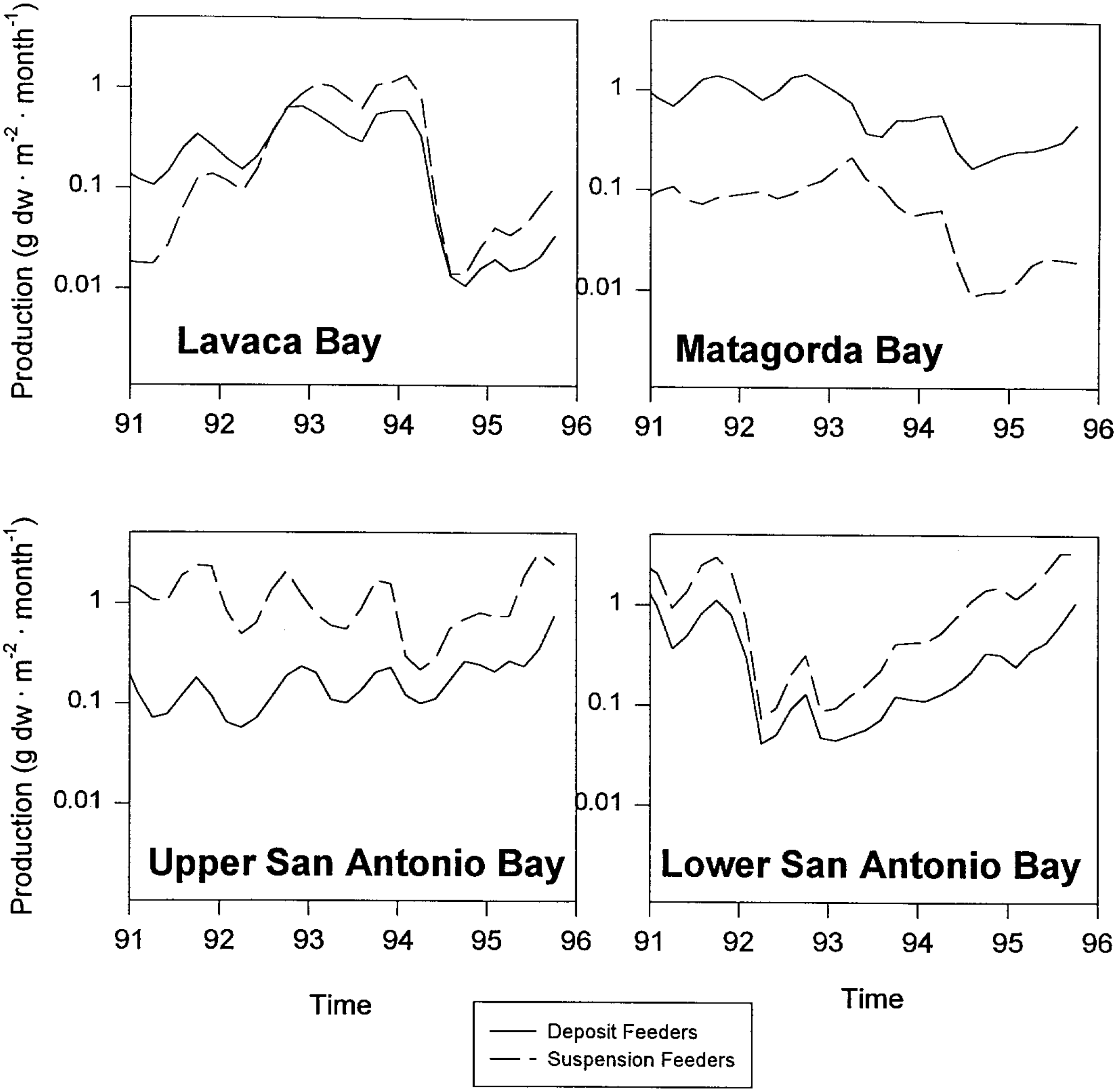


Fig. 3.2.A. The simulated monthly production rate for deposit-feeders and suspension-feeders in the Lavaca-Colorado and Guadalupe Estuaries from 1991 to 1995.

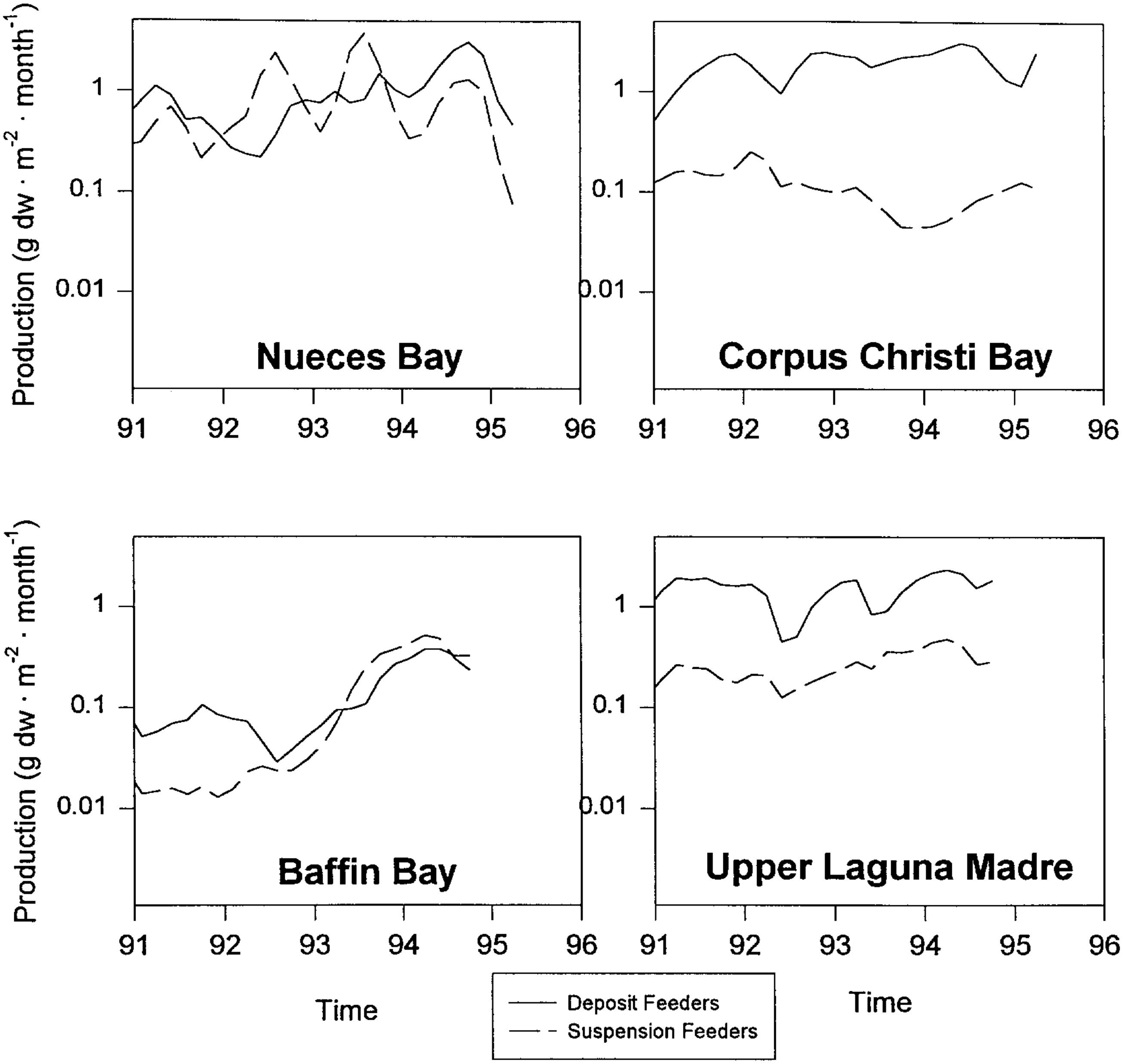


Fig. 3.2.B. The simulated monthly production rate for deposit-feeders and suspension-feeders in the Nueces and Laguna Madre Estuaries from 1991 to 1995.

3.4. Turnover Rate (Annual P/B)

The annual P/B ratio is the annual turnover rate of the benthos, because the units are y^{-1}. The P/B ratios of benthos in the Texas estuaries ranges from 1.2 - 3.2 y^{-1} during the period 1991-1995 (Table 3.1). The P/B ratios for the two trophic groups were similar, ranging from 1.4 - 3.3 y^{-1} for deposit feeders and 2.0 - 3.0 y^{-1} for suspension feeders (Table 3.1). However, during model calibration, the maximal annual P/B ratio for deposit feeders and suspension feeders are estimated to be as high as 5.1 - 6.7 y^{-1} and 5.2 - 8.0 y^{-1} respectively (Table 2.8). The decrease of the annual P/B ratio for both feeding types from higher levels in the calibration run to a lower levels in the simulation run is due to the environmental effects or the environmental limitation over time.

Table 3.1. The average levels of simulated biomass (standing stock), production, and annual P/B through the period from 1991 to 1995. Abbreviations as in Table 2.8.

Bays	Standing Stocks ($g\ dw \cdot m^{-2}$)			Production ($g\ dw \cdot m^{-2} \cdot y^{-1}$)			Annual P/B		
	Deposit feeders	Suspension feeders	Total	Deposit feeders	Suspension feeders	Total	Deposit feeders	Suspension feeders	Total
LB	2.2	1.7	3.9	3.0	4.4	7.4	1.4	2.6	1.9
MB	5.0	0.4	5.4	8.4	0.9	9.3	1.7	2.0	1.7
US	1.4	5.2	6.6	2.2	14.4	16.6	1.6	2.8	2.5
LS	2.1	5.6	7.7	4.4	13.8	18.2	2.1	2.5	2.4
NB	4.4	4.7	9.1	11.7	11.0	22.7	2.7	2.4	2.5
CC	11.1	0.7	11.8	22.9	1.4	24.3	2.1	2.0	1.2
BB	0.6	0.6	1.2	1.7	1.7	3.4	3.0	3.0	2.8
LM	5.5	1.2	6.7	18.3	3.2	21.5	3.3	2.7	3.2

3.4. Temporal Variation of Monthly Production Rate and Environmental Limitation

The annual P/B ratios that are listed in Table 3.1 are the average levels for the production rate of benthic biomass. During the period 1991-1995, the monthly production rates (i.e. monthly P/B) changes seasonally based on the environmental limitations. A comparison between the monthly P/B and the environmental limitation factors illustrates why there is temporal variation in the production values. Variation of production is different from variation of biomass due to fish predation. A high monthly P/B indicates that the benthos biomass is able to recover standing stock levels rapidly following consumption by fish. A low monthly P/B means that the standing stock falls to a lower level and doesn't recover following predation mortality.

The variance of monthly P/B ratios of both feeding types was different in all eight bays from 1991 to 1995 (Fig. 3.3). Overall, the suspension feeders had higher variance than deposit feeders. Deposit feeders in the primary bays close to the sea (Matagorda Bay, Lower San Antonio Bay, Corpus Christi Bay and Laguna Madre) had higher variation than in the secondary bays close to freshwater sources (Lavaca Bay, Upper San Antonio Bay, Nueces Bay and Baffin Bay). The opposite is true for suspension feeders. Specially, in the Upper San Antonio Bay, suspension feeders are the most variable, while deposit-feeders are the most stable relative to other bays.

The temporal variation of the monthly P/B correlated with the variation of environmental limitation factors (Figs. 3.4.A-D). Temperature is a basic environmental limitation that affects physiological rates. It affects the production rate of both feeding types in each bay with minor seasonal fluctuation. However, there is not a large difference in temperature limitation between feeding types or among eight bays. Salinity is the major cause of environmental limitation. Salinity variation has a greater affect on the production rate than does temperature. Deposit feeders have higher salinity limitation than suspension feeders in secondary bays close to freshwater inflow sources, e.g., in the Lavaca-Colorado Estuary (Fig. 3.4.A), Guadalupe Estuary (Fig. 3.4.B) and Nueces Estuary (Fig. 3.4.C). Suspension feeders had higher salinity limitation in the primary bays close to the sea, e.g., in the Lavaca-Colorado Estuary (Fig. 3.4.A), Nueces Estuary (Fig. 3.4.C), and Laguna Madre (Fig. 3.4.D). Finally, food availability had the least limitation effect on both feeding types in all eight bays. The food limitation factor was close to 1 for all bays, indicating little or no effect due to food limitation (Figs. 3.4.A-D).

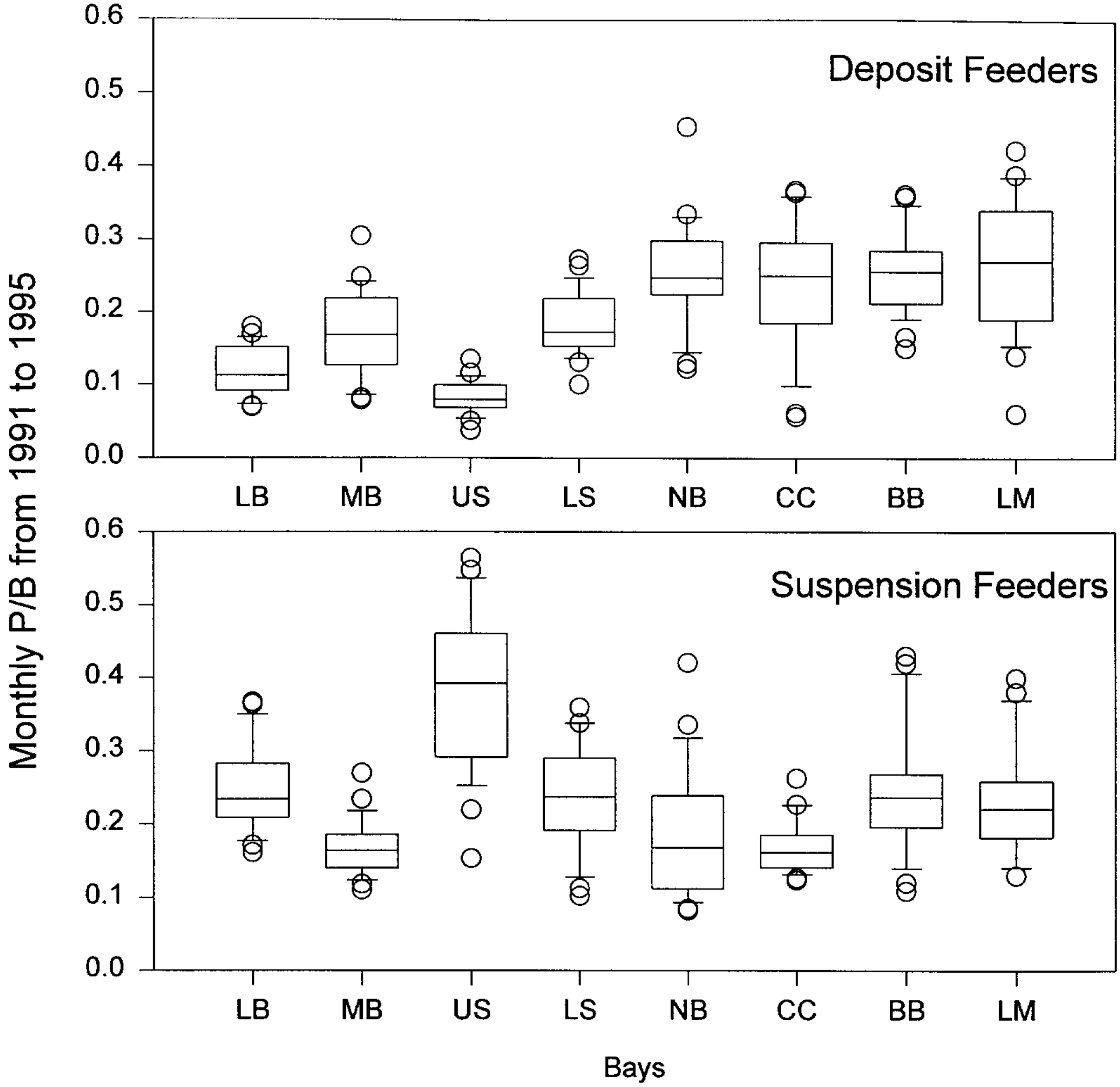


Fig. 3.3. Tukey Box plots for the distribution of the monthly P/B for two feeding types in eight bays. The box encompasses the 25th through 50th percentile points of the data, the horizontal lines mark the 10th , 50th , and 90th percentiles, and the symbols mark the 5th and 95th percentiles.

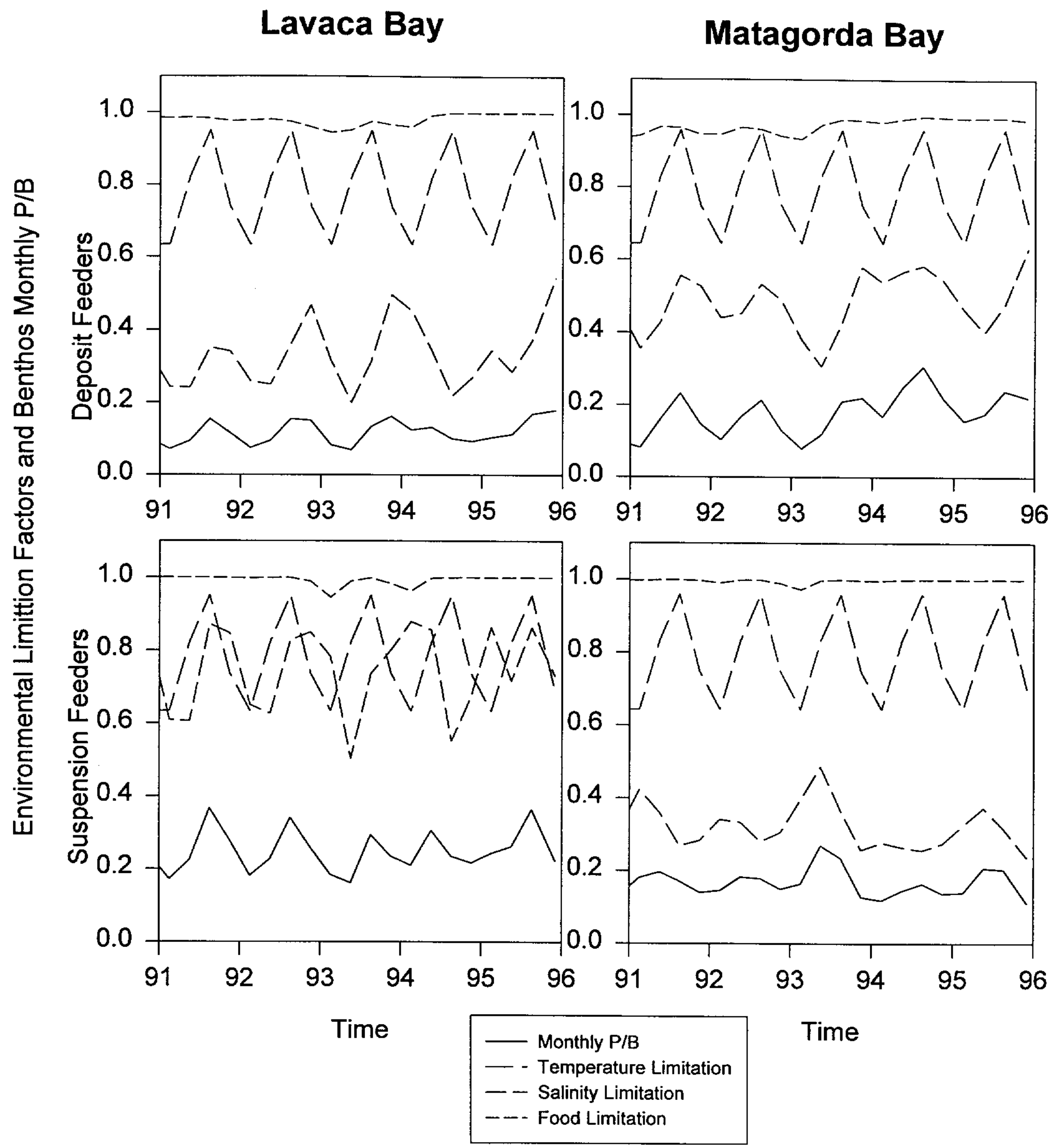


Fig. 3.4.A. Environmental limitation and simulated monthly production rates of two feeding types in the Lavaca-Colorado Estuary. The environmental limitation includes temperature, salinity and food availability limitation where 1=no limitation and 0.5=growth rate limited to 50% of the maximal rate.

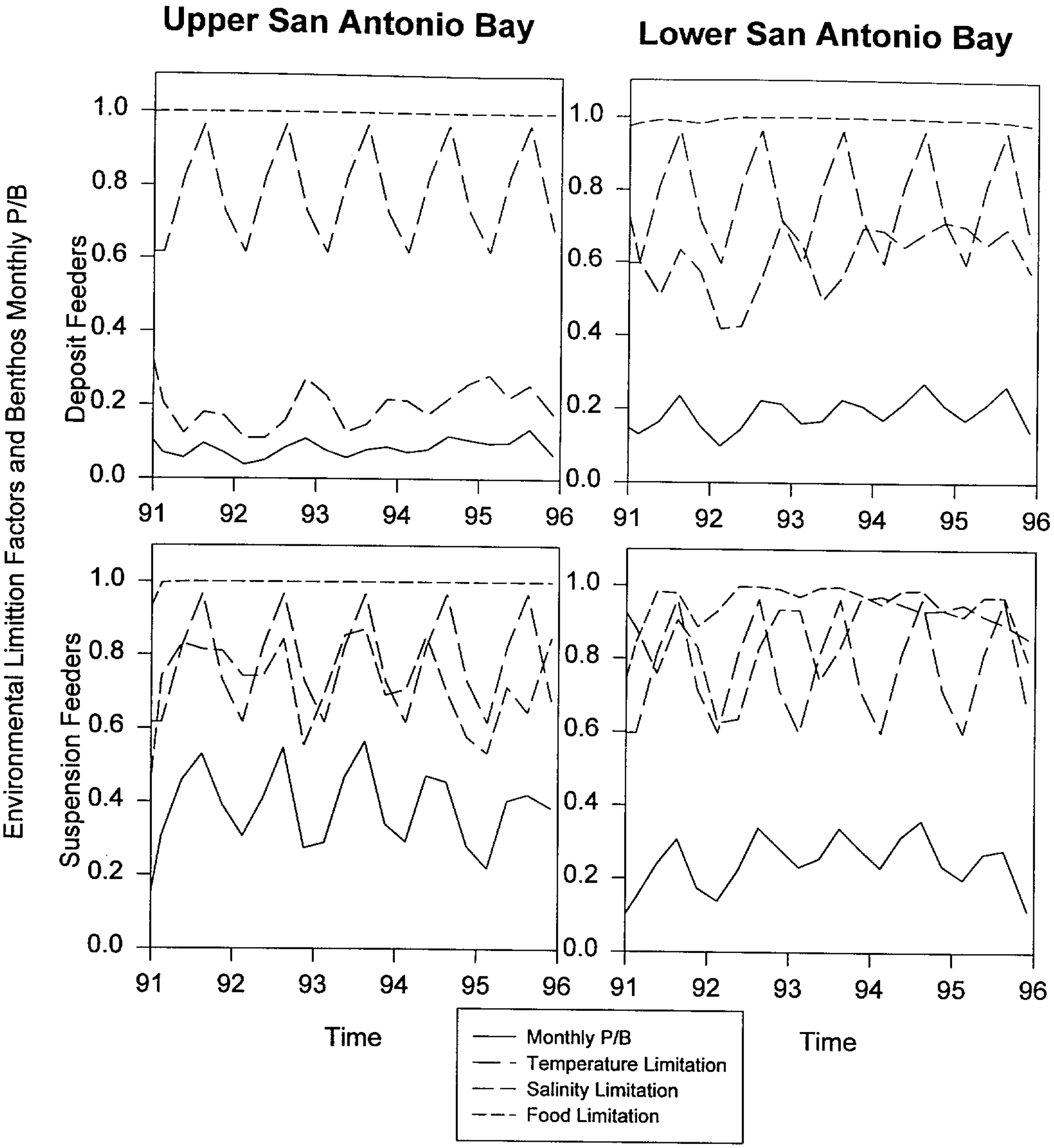


Fig. 3.4.B. Environmental limitation and simulated monthly production rates of two feeding types in the Guadalupe Estuary. The environmental limitation includes temperature, salinity and food availability limitation where 1=no limitation and 0.5=growth rate limited to 50% of the maximal rate.

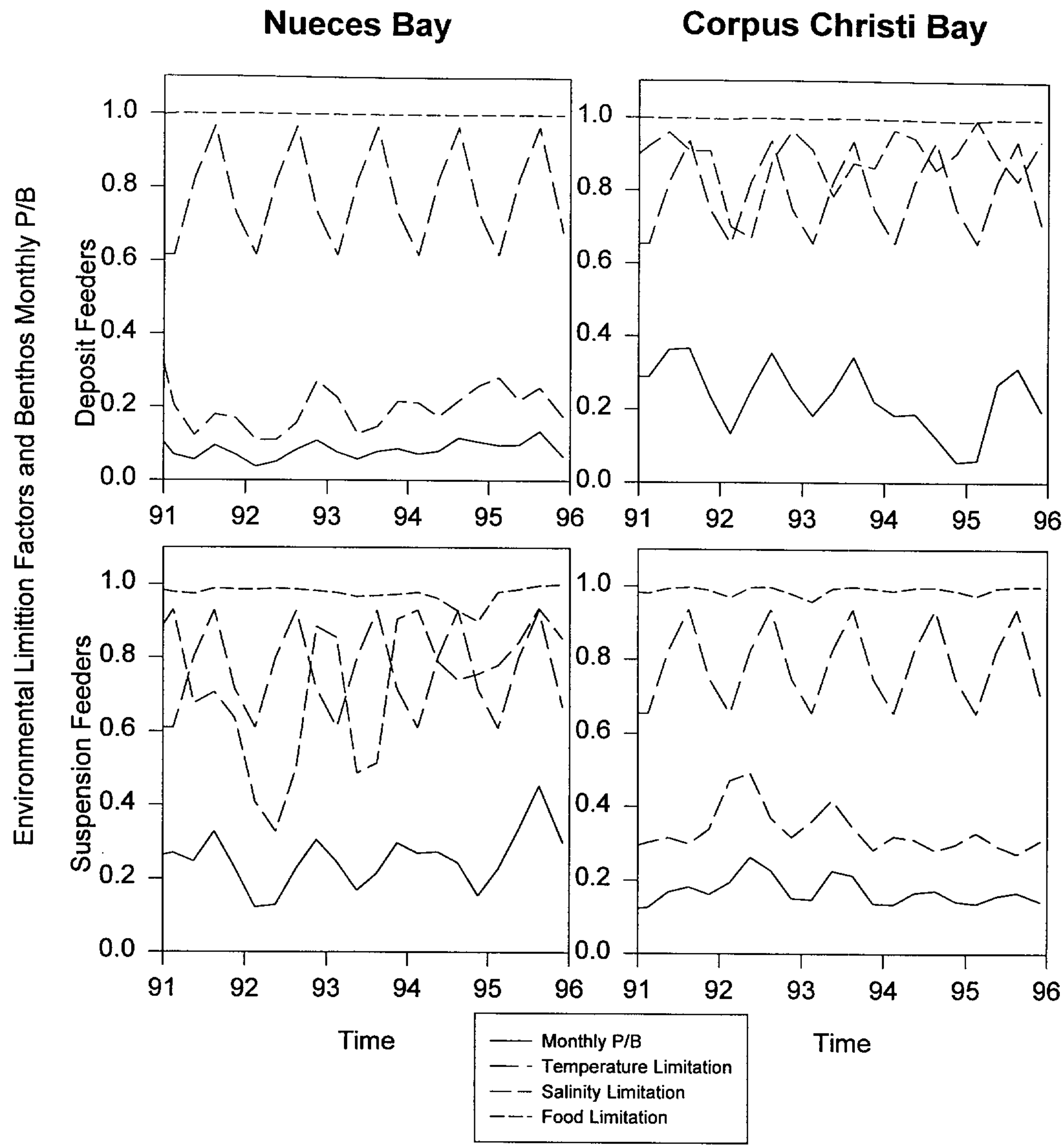


Fig. 3.4.C. Environmental limitation and simulated monthly production rates of two feeding types in the Nueces Estuary. The environmental limitation includes temperature, salinity and food availability limitation where 1=no limitation and 0.5=growth rate limited to 50% of the maximal rate.

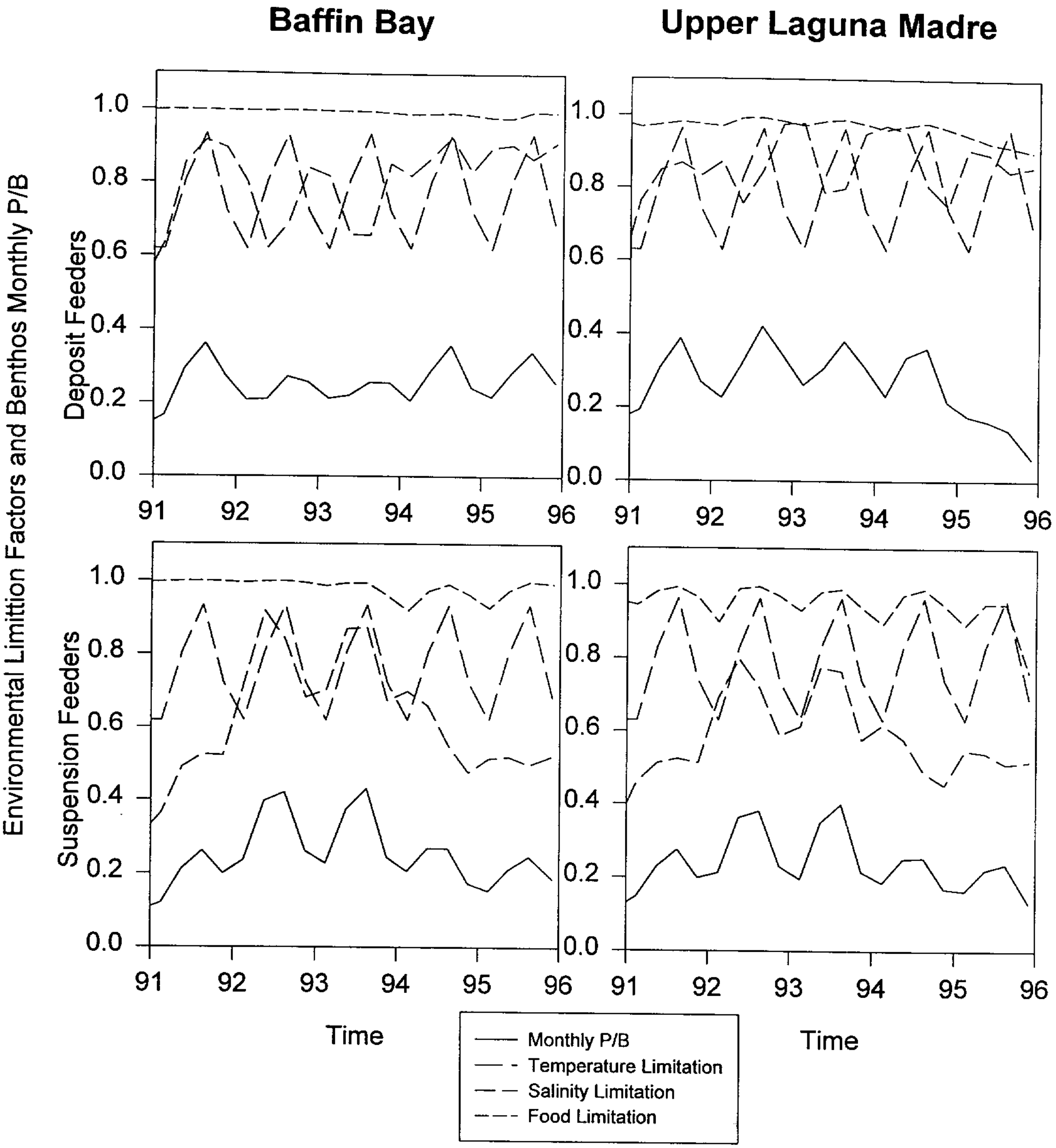


Fig. 3.4.D. Environmental limitation and simulated monthly production rates of two feeding types in the Laguna Madre Estuary. The environmental limitation includes temperature, salinity and food availability limitation where 1=no limitation and 0.5=growth rate limited to 50% of the maximal rate.

3.5. Biomass Loss By Fish Predation

The variation of benthic biomass can be affected by the variation of fish predation rates. The variation of predation mortality caused by fish ranges from zero to more than 15 g dw · m^{-2} · $month^{-1}$ (Fig. 3.5). There is a low predation rate of fish on deposit feeders in Lavaca Bay, Lower San Antonio Bay, Upper San Antonio Bay, and Baffin Bay, and on suspension feeders in Lavaca Bay, Corpus Christi Bay, Baffin Bay and Laguna Madre. High rates of predations on both groups occur in Matagorda Bay and Nueces Bay only (Fig 3.5).

Fluctuations of biomass following changes in predation rates is evident (Figs. 3.6.A-D). A strong reduction of benthos biomass occurs after the peaks of predation mortality, such is the case for both feeding types in Lavaca Bay in 1994, Lower San Antonio Bay in 1992, and Corpus Christi Bay in 1992.

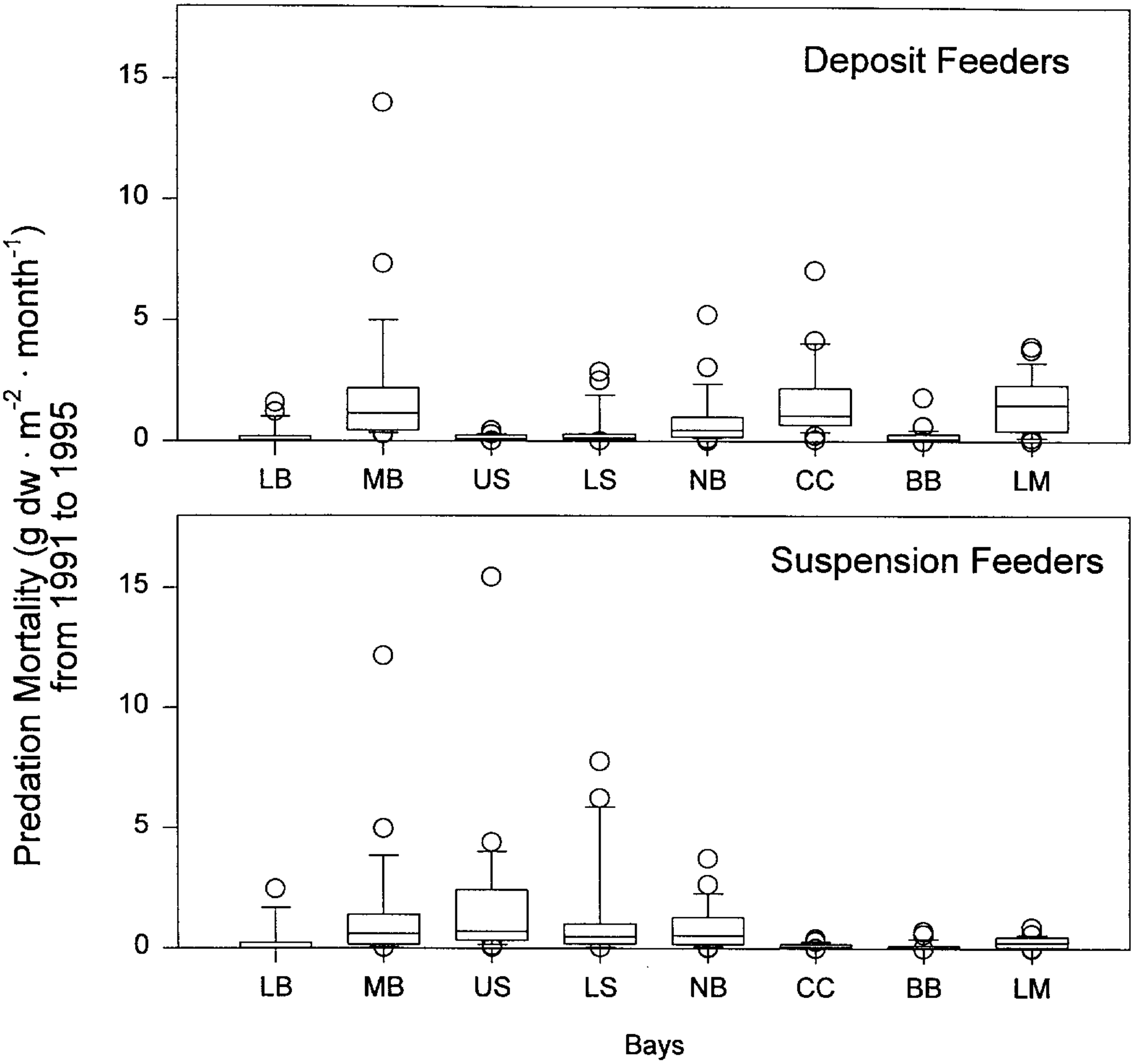


Fig. 3.5. Tukey Box plots for the distribution of the monthly predation rates on two benthos feeding types in eight bays. The predators include: red drum, black drum and blue crab.

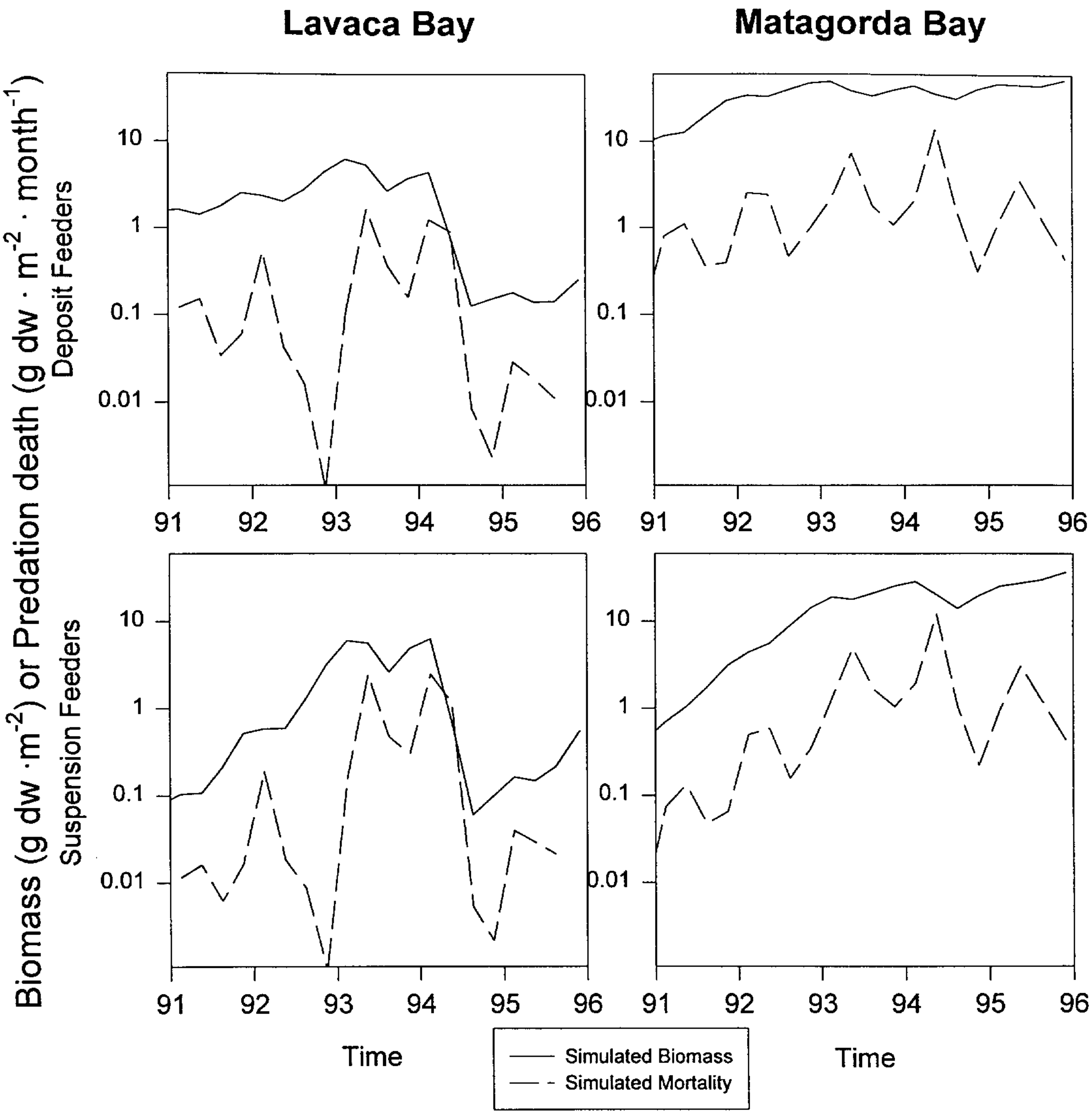


Fig. 3.6.A. Simulated monthly predation rate on two benthic feeding types in Lavaca-Colorado Estuary.

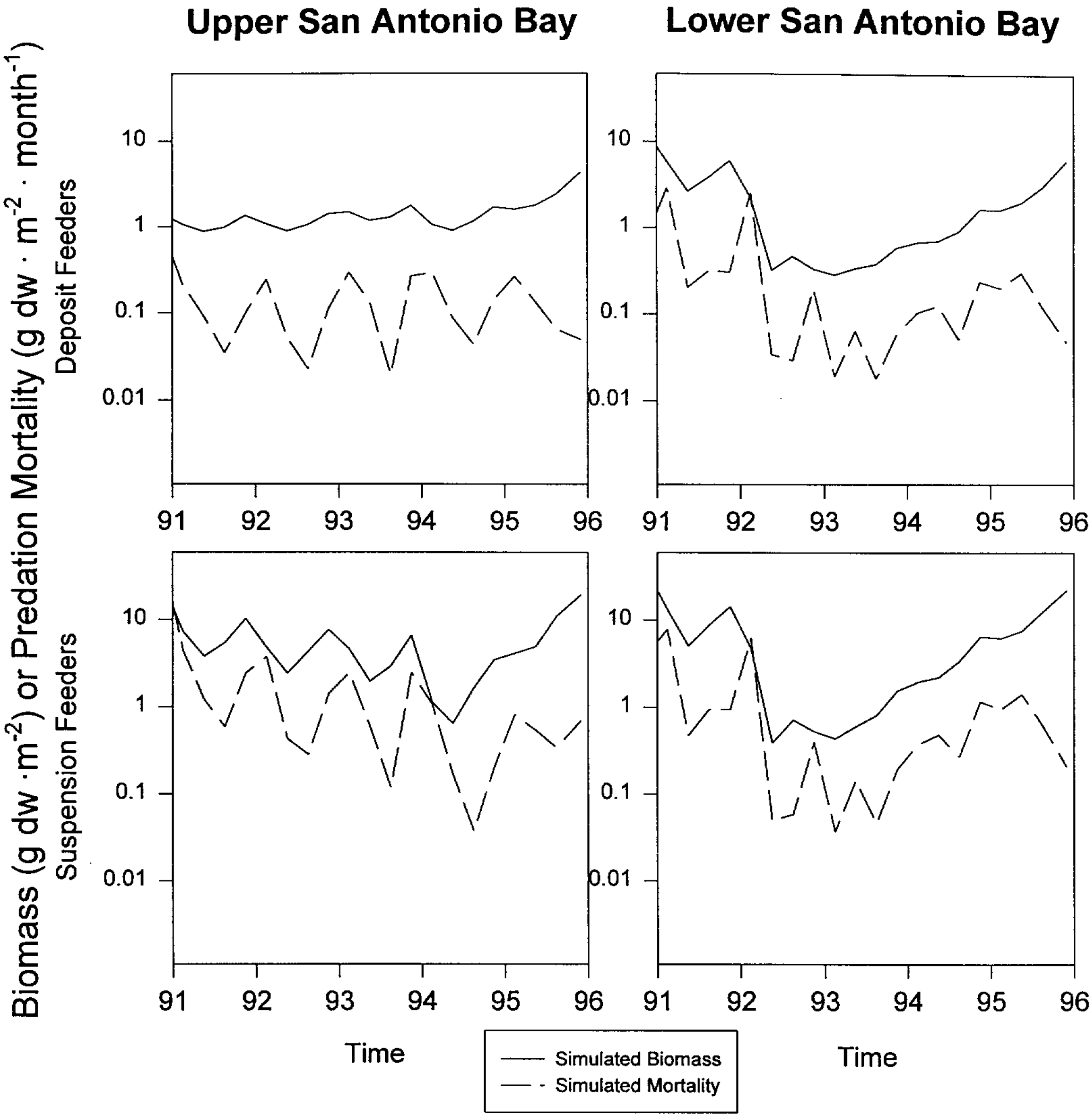


Fig. 3.6.B. Simulated monthly predation rate on two benthic feeding types in Guadalupe Estuary.

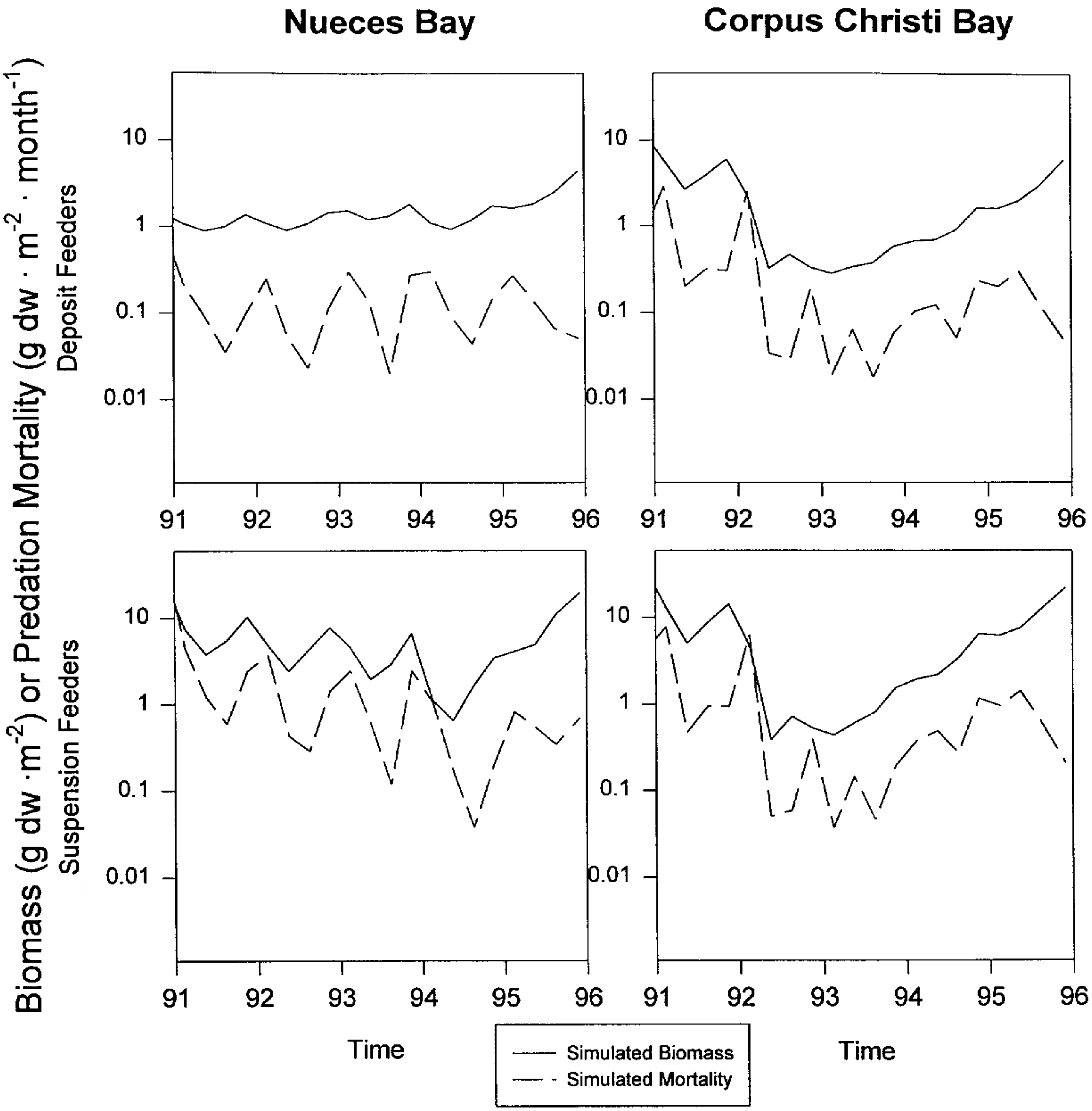


Fig. 3.6.C. Simulated monthly predation rate on two benthic feeding types in Nueces Estuary.

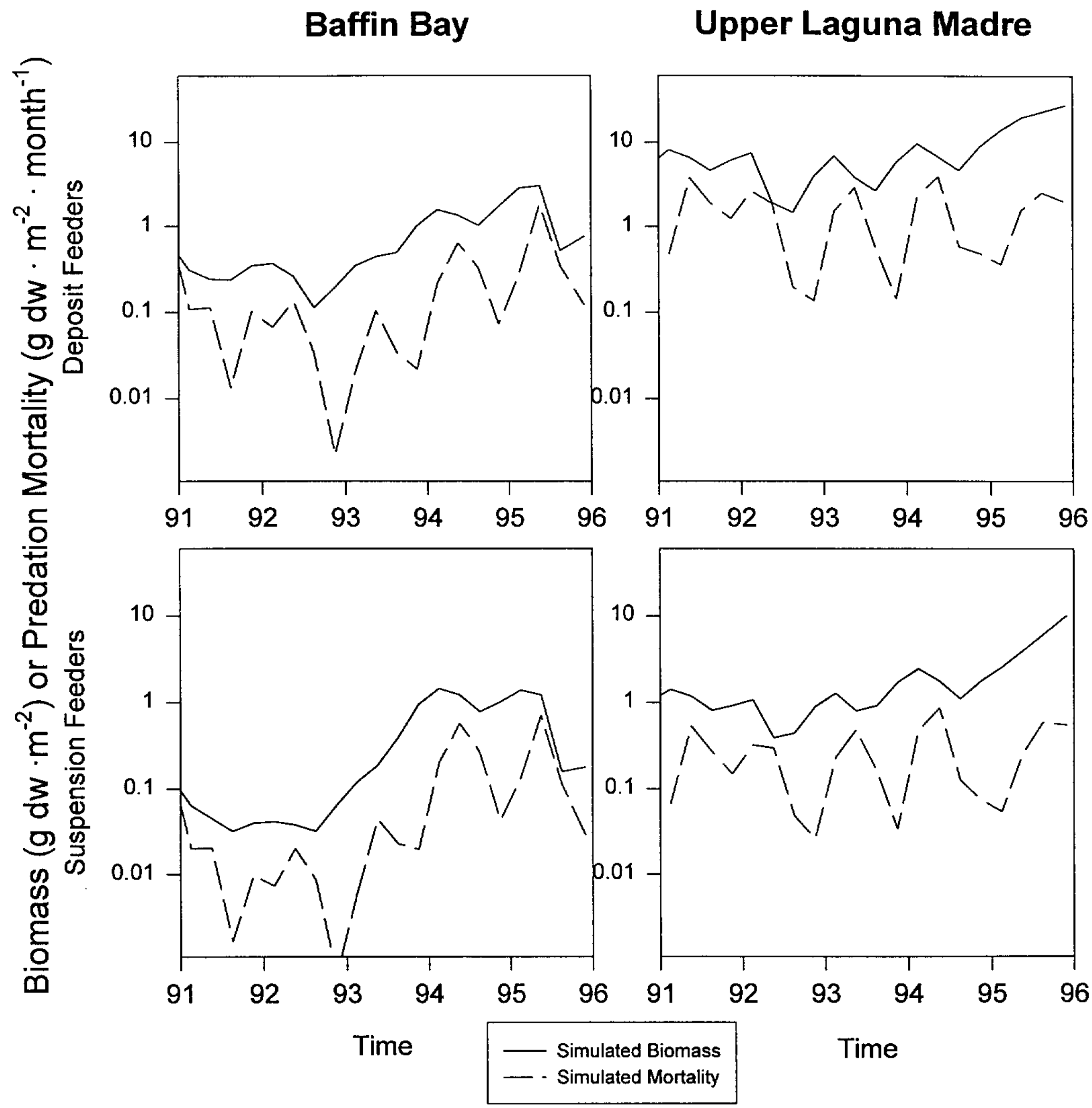


Fig. 3.6.D. Simulated monthly predation rate on two benthic feeding types in Laguna Madre Estuary.

4. DISCUSSION

A bio-energetic model, calibrated using a long-term data set on benthic biomass, was developed that relates macrobenthic productivity to salinity within and among four Texas estuaries, the Lavaca-Colorado, Guadalupe, Nueces and Laguna Madre. Benthic biomass data, which is an indicator of secondary productivity (Banse and Mosher, 1980), was available for a 5-year period (1990-1995) in all estuaries, and over a longer period (from 1988) in some estuaries. Biomass was computed for two trophic groups: deposit feeders (that consume detritus or sediment organic matter) and suspension feeders (that filter phytoplankton or graze on benthic diatoms). The bio-energetic model was based on biological processes. Food inputs to macrofauna included sedimentary POM for detritivores and modeled primary production for grazers. Sediment POM was measured, and the model for primary production was based on measured nutrient concentrations and light. Limited information was available for the range of phytoplankton standing stocks or rates of primary production. Organismal limitation was modeled for salinity, temperature, and predation by higher trophic levels. The main predators on benthic infauna were assumed to be red drum, black drum, and blue crabs.

4.1 Main Conclusions

Models of the estuaries as a whole were not successful at simulating the observed biomass. Instead, when each estuary was divided into a primary and secondary bay, then the model was more successful at predicting macrofaunal biomass. The final model run was based on eight Texas bays, two within each of the four estuaries. The better fit of the data to individual bay models indicates that strong gradients exist within Texas estuaries. These gradients are the direct result of freshwater inflow influencing the secondary bay, and marine exchange influencing primary bays. This difference among bays is a general feature of Texas estuaries.

Within estuaries the average annual production to biomass ratio (P/B), which is in units of y^{-1}, increased with proximity to the freshwater inflow source (Table 3.1). The range of P/B values indicates that macrofauna in Texas Bays turns over 1 to 3 times per year. Low turnover times are associated with bays that have low inflow rates and other anthropogenic disturbances. Corpus Christi Bay had the lowest P/B of 1.2 y^{-1}, and this is probably due to several forms of natural and anthropogenic disturbance, including low inflow, poor circulation, disturbance due to shrimp trawling, and extensive marine development. The one exception was Laguna Madre, with the highest P/B of 3.2 y^{-1}. This high rate is associated with extensive seagrass habitat, so turnover rates are also enhanced by the habitats present within the estuaries.

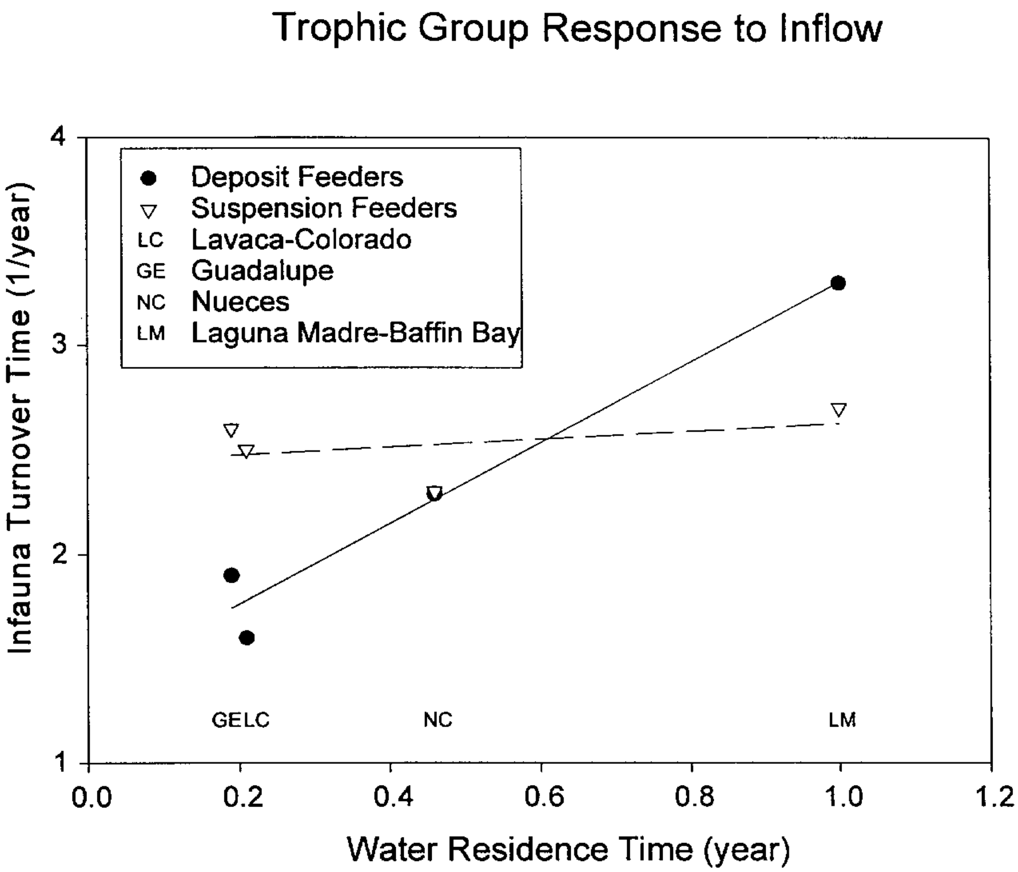


Fig. 4.1. Trophic group response in terms of P/B (y^{-1}) to freshwater residence time (y). Each point represents the measurement from one estuary, which is placed on the abscissa at its water residence time.

Estuarine physiography is also important, because salinity change is a function of inflow being diluted by the volume of the estuary. The estuarine water residence time is calculated by the fraction of freshwater diluted by oceanic water: $f \times (V \div Q)$; where $f =$ (bay salinity - Gulf salinity) $\div$ Gulf salinity, V = volume of the estuary, and Q = the net inflow (Armstrong, 1982). Residence time is calculated on an estuarine wide basis, so the standing stocks and productivity for each estuary was calculated by adding the values of the estuarine components from Table 3.1. The P/B ratio for deposit feeders increases with water residence time (Fig. 4.1, r=0.98, P=0.018). This is likely due to the fact that as water residence time increases, the environment is increasing depositional. Therefore, deposit feeders are enhanced. In contrast, we would predict that suspension feeder turnover times would increase as water turnover time increases, which means residence time decreases. On first inspection, it appears that this is not the case (Fig. 4.1, r=0.42, P=0.58). Laguna Madre has a high turnover time and a long residence time. However, Laguna Madre benthos is dominated by a seagrass fauna, which is very dynamic due to the submerged aquatic habitat. When only the bays with the soft-bottom benthic habitat are included (GE, LC, and NC), the trend is negative as predicted, but not statistically significant due to the low sample size (Fig. 4.1, r=-0.96, P=0.17). There are two implications to this finding. The first is that suspension feeders, particularly mollusks are the best indicators of the importance of freshwater inflow to maintaining the productivity of an ecosystem. This is consistent with previous findings (Montagna and Kalke, 1995). The second implication is that communities in bays with inflow that is reduced artificially will change in dominance patterns, from a suspension feeder community dominating to a deposit feeder community dominating. This change from a mollusk and crustacean dominated community to a polychaete dominated community is consistent with an emerging paradigm on how benthos changes as a result of anthropogenic disturbance (Peterson et al., 1996).

4.2. Consistency With Hypotheses From the Conceptual Model

In the conceptual modeling study (Montagna et al., 1995), we predicted that Laguna Madre would have a higher rate of energy flow passing through the system. This hypothesis is based on two characteristics unique to Laguna Madre: its large size and the extensive seagrass habitat. In this quantitative modeling study, we estimated that the benthos of Laguna Madre has the highest annual P/B at 3.2 y^{-1} in comparison with other bays which range from 1.2 - 2.5 y^{-1}. The higher production is mainly due to production of the deposit-feeders, which is up to 18.3 g dw $\cdot m^{-2} \cdot y^{-1}$. These deposit feeders are probably exploiting a detrital foodweb enhanced by seagrass detritus.

In the conceptual model, we predicted that higher inflow would yield high productivity of suspension feeders, because river-born nutrients would stimulate primary production, which in turn would fuel benthic productivity. San Antonio Bay, has the highest levels of nutrients of all bays, and the highest level production of suspension feeders in all bays. Suspension feeder production in San Antonio Bay is 14 g dw $\cdot m^{-2} \cdot y^{-1}$ in comparison with a range of 0.9 - 11 g dw $\cdot m^{-2} \cdot y^{-1}$ found in other bays. However, the annual P/B ratio in San Antonio Bay, which is 2.5 y^{1}, is not highest in comparison to the range of other Texas bays 2.0 - 3.0 y^{-1}.

4.3 Problems With the Model

As might be expected, modeling exercises do not always yield perfect results. In general, one can expect to encounter at least two main problems: a lack of realism or completeness of the model, and a lack of data to calibrate the model. While these problems definitely occurred, there were two other interesting problems, the lack of consistency when modeling different periods, and the poor fit of any model with data from Lavaca Bay.

4.3.1. Contrasting the Short Period and Long Period Simulations

One odd result is that the simulation for the longer period of time (1988 - 1995) does not fit the data better that the simulation for the short-term period (1990 - 1995). Some differences do occur between the results of the short- and long-period simulations. The simulation of the long-period (1988-1995) predicts higher standing stock levels and maximal annual P/B levels (Table 2.9). This may due to the fact that there was higher biomass in most bays prior to 1991. There are several reasons why the short and long term simulation could give different results. The most obvious reason is that the simulation is not complete. There are other environmental effects and state variables that have not yet been considered or incorporated into the model. Therefore, the simulation error increases simply because there are more observations over a longer period of time that have to fit the model. A second possibility is that the long-term database used was not complete. We were missing nutrient data prior to 1991 for Lavaca-Colorado and Laguna Madre Estuaries, and during 1989-1990 for Guadalupe and Nueces Estuaries. The third possibility is that it is simply more difficult to estimate the parameters in the calibration run for long-term period than for the short-term period. We have done about 10,000 calibration runs for both simulation periods. However, it is possible that many more runs are necessary for long-term simulation, because of the increase in total observation error.

4.3.2. The Worst-fit Simulation: Lavaca Bay

The worst fitting simulation for both deposit-feeders and suspension-feeders biomass occurred in Lavaca Bay. This may be due to one of the two the following reasons. There was no observational trawl data available for red drum and black drum for the simulation period from the Texas Park and Wildlife Department for this Bay. These predator fish exist in the Bay, but haven't been recorded. Predation mortality was very important in limiting biomass (Figs. 3.6.A-D). This lack of data means that the simulation of predation mortality is under estimated, therefore biomass was generally overestimated (Fig. 3.1.A). The second reason is that Lavaca Bay may have a high level of pollutant input that can affect benthos productivity, and pollutants are not considered in this study. Lavaca Bay is currently a U.S. EPA Superfund site with high levels of mercury contamination.

The P/B ratio based on the current simulation is very low, in the range of 1.7-1.9 y^{1}. This value is in the lower range of all the Texas Bays. Productivity is probably much higher, because it is being consumed by fish. Unfortunately, we don't have sufficient data to get a better estimate of productivity at this time.

4.4. Improving Model Structure and Performance

All models can be improved. The two most obvious improvements are a longer term data set and the inclusions of processes not in the current model. One obvious problem is that the period for the best data (1990-1995) is short, and covers only one wet to dry cycle. Generally, a minimum of two cycles should be covered to have a good model fit.

4.4.1 Need for Complete Data Sets

Several obvious directions should be taken in the future, but a lack of long-term synoptic data is the most overwhelming problem. We are beginning to obtain a very good data set on benthic biomass at the UTMSI, and the Texas Parks and Wildlife Department, Coastal Fisheries Division has an excellent long-term data set on finfish and large shellfish. We have only spotty amounts of data for nutrients, chlorophyll, and sediment organic matter.

4.4.2 Using Salinity as a Surrogate for Inflow

The current model did not attempt to use freshwater inflow as the forcing function for biological activity. Instead, observations of salinity and nutrients were used as the forcing functions, and salinity is used as a surrogate for inflow. This may introduce error into the model or may be too indirect for water management purposes. The best way to avoid these errors is to develop a physical model using freshwater inflow as the forcing function to predict salinity and nutrients. The simulated salinity and nutrients could be used as input to the biological model. Due to the complex nature of the interactions between freshwater and seawater mixing and movement, the physical model would have to simulate the surface inflow and inflow balance:

$$\text{Surface Inflow} = \text{Gaged} + \text{Model} + \text{Return} - \text{Diversion}$$

$$\text{Inflow Balance} = \text{Surface Inflow} + \text{Precipitation} - \text{Evaporation} - \text{Diversion}$$

The simulation of Nueces and Laguna Madre Estuaries are much better than the simulations of Lavaca-Colorado and Guadalupe Estuaries. There are two implications to this result. First, further consideration of pollution effects on the limitation of the benthos productivity may be needed. Second, the model fits estuaries with low inflow (Nueces and Laguna Madre Estuaries) better than estuaries with high inflow (Lavaca-Colorado and Guadalupe Estuaries). This may be due to a larger affect of interannual variability of inflow in the high inflow estuaries than in the low inflow estuaries. The estuaries with low inflow generally have a smaller salinity range from year-to-year than the high inflow estuaries. This could be causing the observed biological effects.

4.4.3 Modeling Predation Effects.

During model development, we found that it was difficult to model and collect data on predation effects. In the beginning, we used the sum of Nemertinea, as an infaunal predator, and commercially harvested fish. However, predation pressure to benthos may be different between fish and invertebrate predators, and the fish data was only available for estuarine-wide annual averages, so the spatial and temporal scales were very coarse. This led to very poor simulations. We then tried using the Coastal Fisheries monitoring data for red drum, black drum, catfish and flounder (McEachron & Fuls, 1996). This data was also annual data for the entire estuarine system. This simulation was also very poor when compared to observations of benthic biomass. Finally, we obtained the actual field collection notes from Coastal Fisheries, and we were able to estimate the amount of fish in a given bay for each month of the year. This data set gave us the best simulations. Interestingly, the worst simulations is for Lavaca Bay, where some of the fisheries data was missing.

4.4.4 Modeling Multiple Stressors

For most cases, a decrease in benthos biomass is expected due to predation. However, in a case where there are multiple environmental stressors (such as the presence of pollution and unsuitable salinity ranges), respiration and natural mortality could be higher than other losses or growth. In this case, benthos biomass may be lost before predation pressure can have an effect, and there may even be a net negative growth rate. The current limitation equation (5) assumes that the minimal growth rate is a function of just salinity, temperature, and food, and can only be as low as zero. To set up a negative growth rate, we can transform all environmental limitations from the range between 0 and 1 to a new range between -1 and 1. An additional parameter, $p_{(ng)}$, is included to model the case that includes negative growth and positive growth. The new environmental limitation become:

$$E_{ben}=(p_{(ng)}+1)\cdot(E_{sal}+E_{tem}+E_{food})-p_{(ng)} \qquad \textbf{(16)}$$

where $p_{(ng)}$ is new parameter to be calibrated. It has range from 0 to 1. When $p_{(ng)}=0$, there is no negative growth and equation (8) become equations (7). When $p_{(ng)}=1$, the maximal environmental limitation is occurring and this forces benthos biomass to decrease before predation even occurs.

A good example of the use of this term would by for Laguna Madre. The model has not yet considered the affect of brown tide that may cause a decline in benthos (Conley et al., 1997). This could be included in the model. Another example, is modeling the affect of mercury contamination in Lavaca Bay. Inclusion of the multiple stressor term could improve model performance, especially in Lavaca Bay.

4.5 Summary

A bio-energetic model was developed that relates macrobenthic productivity to salinity differences within and among eight Texas bays. The model was developed for four Texas estuaries, the Lavaca-Colorado, Guadalupe, Nueces and Laguna Madre, that lie in a climatic gradient with decreasing rainfall and concordant decreasing freshwater inflow. A long-term data set on macrobenthic biomass, which is an indicator of productivity, was used to calibrate the model. The benthos were divided into two trophic groups: deposit feeders (that consume detritus or sediment organic matter) and suspension feeders (that filter phytoplankton or graze on benthic diatoms). Within estuaries, the P/B increased with proximity to the freshwater inflow sources. The P/B ratio of deposit feeders generally increased with water residence time, i.e., inflow volume adjusted by the estuary volume. In contrast, the P/B ratio for suspension feeders decreased with water residence time, except for Laguna Madre where there is vegetated habitat. The different response between the two feeding groups implies that suspension feeders are the best indicators of inflow effects, and altered inflows will cause community structure changes. Overall, Laguna Madre had the highest P/B of 3.2 y^{1}, which is due to extensive seagrass habitat. Corpus Christi Bay had the lowest P/B of 1.2 y^{1}, and this is probably due to several forms of natural and anthropogenic disturbance, which includes very low inflow rates.

6. REFERENCES

Armstrong, N. E., 1982. Responses of Texas Estuaries to freshwater inflows. In Kennedy, V. S. ed., *Estuarine Comparisons,* Academic Press, p 103-120.

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 (7.12), pp. 105.

Banse, K. and S. Mosher. 1980. Adult body mass and annual production/biomass relationships of field populations. *Ecol. Monogr.* 50:355-379.

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. John Wiley &Sons, 688 pp.

Carrada G. C. 1983. Modeling of the Gulf of Naples, In: Quantitative analysis and simulation of Mediterranean coastal ecosystems: The Gulf of Naples, a case study (eds.: G. Carrada, T. Hopkins, L. Jeftie and S. Morcos), *UNESCO Reports in Marine Science* 20: 80-153.

Conley, M.F., Montagna, P.A., and Kalke, R.D. 1997. Effect of a persistent brown tide bloom on macroinfaunal communities in Baffin Bay and Laguna Madre, Texas. In Prep.

de Hoop, B. J., Herman, P. M. J., Scholten, H. and Soetaert, K., 1989 - SENECA 1.5: A Simulation Environment for Ecological Application. Netherlands Institute of Ecology, Center for Estuarine and Coastal Ecology, 180 pp.

Holt, S. A. and Arnold, C. R., 1989. Nursery habitat utilization by finfish and shellfish in Lavaca, San Antonio, and Mesquite Bay and their relationships to freshwater inflow. *Final Report, Data Synthesis Study (NIPS), The University of Texas Marine Science Institute, Technical Report No. TR/89-009.*

Keen R. E. and J. D. Spain. 1992. Computer simulation in biology: a BASIC introduction. Wiley-Liss, New York, 498 pp.

Larkin, T. J. and Bomar, G. W., 1983. Climatic atlas of Texas. Texas Department of Water Resources. Austin, Texas, 151 pp.

Li, J., Vincx, M. and Hermen, P. M. J. 1996. A model for nematode dynamics in the Westerschelde estuary. *Ecological Modeling*, 90: 271-284.

Lotka, A. J. 1925. Elements of physical biology. Williams & Wilkins, Baltimore, Md, 460 pp.

Kalke, R.D. and 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*, 32:49-72.

Mannino, A. and Montagna, P.A. 1996. Small scale spatial variation of macrobenthic community structure. *Estuaries* In Press.

Martin, C.M. and Montagna, P.A. 1995. Environmental assessment of La Quinta Channel, Corpus Christi Bay, Texas. *Texas J. Sci.* 47:203-222.

McEachron, L. W. and Fuls, B., 1996. Trends in relative abundance and size of selected finfishes and shellfishes along the Texas coast, Coastal Fisheries Division, Management Data Series, in press.

Montagna, P.A. 1989. Nitrogen Process Studies (NIPS): the effect of freshwater inflow on benthos communities and dynamics. Technical Report No. TR/89-011, Marine Science Institute, The University of Texas, Port Aransas, TX, 370 pp..

Montagna, P.A. 1991. Predicting long-term effects of freshwater inflow on macrobenthos in the Lavaca-Colorado and Guadalupe Estuaries. Final Report to Texas Water Development Board. Technical Report No. TR/91-004, Marine Science Institute, The University of Texas, Port Aransas, TX, 78 pp.

Montagna, P.A. 1994. Inflow needs assessment: Effect of the Colorado River diversion on benthic communities. Final Report to the Lower Colorado River Authority. Technical Report No. TR/94-001, Marine Science Institute, The University of Texas, Port Aransas, TX. 63 pp.

Montagna, P.A. and R.D. Kalke. 1992. The Effect of freshwater inflow on meiofaunal and macrofaunal populations in the Guadalupe and Nueces Estuaries, Texas. *Estuaries*, 15:266-285.

Montagna, P.A. and R.D. Kalke. 1995. The ecology of infaunal Mollusca in south Texas estuaries. *Amer. Malac. Bull.* 11:163-175.

Montagna, P.A. and R.D. Kathmann. 1996. Bioaccumulation of mercury in Lavaca Bay, Texas assessed by using stable isotopes. In prepartion.

Montagna, P. A., D. A. Stockwell and Kalke, R. D. 1993. Dwarf surfclam *Mulina lateralis* (Say, 1822) populations and feeding during the Texas brown tide event. *Journal of Shellfish Research, 12: 433-442.*

Montagna, P.A. and Yoon, W.B. 1991. The effect of freshwater inflow on meiofaunal consumption of sediment bacteria and microphytobenthos in San Antonio Bay, Texas USA. *Estuarine and Coastal Shelf Science*, 33:529-547.

Odum, H. T. and R. Wilson, 1962. Further studies on reaeration and metabolism of Texas Bays, 1958-1960. *Publ. Inst. Mar. Sci., U.T.*, 8:23-55.

Orlando, S.P., Jr., L.P. Rozas, G. H. Ward, and C. J. Klein, 1993. Salinity characteristics of Gulf of Mexico Estuaries. Silver Springs, MD: National Oceanic and Atmospheric Administration, Office of Ocean Resources Conservation and Assessment. 209 pp.

Peterson, C.H., M.C. Kennicutt II, R. Green, P. Montagna, E. Powell, and P. Roscigno. 1996. Ecological consequences of environmental perturbations associated with offshore hydrocarbon production: A perspective from study of long-term exposures in the Gulf of Mexico. *Canadian Journal of Fisheries and Aquatic Sciences* In Press.

Quast, W. D., Searcy, T. S. and Osborn, H. R. 1988 Trends in Texas commercial fishery landings, 1977-1987. Management Data Services, No. 149. Texas Parks and Wildlife Department, Coastal Fisheries Branch, Austin, Texas, 107 pp.

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.

Tumbiolo, M. L. and Downing, J. A., 1994. An empirical model for the prediction of secondary production in marine benthic invertebrate populations. *Mar. Ecol. Prog. Ser.*, 114: 165-174.

Whitledge, T. E., 1989. Nutrient distributions and dynamics in Lavaca, San Antonio and Nueces/Corpus Christi Bays in relation to freshwater inflow. Final Report, Data Synthesis Study (NIPS), The University of Texas Marine Science Institute, Technical Report No. TR/89-007.

Wohlschlag, D. E., J. M. Wakeman, R. Vetter, and Ilg, R. J. 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, by Marine Science Institute, University of Texas at Austin, Texas, 105 pp.

7. APPENDIX: 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

```
C#######################################################################
C SENECA 2.0                                          (C) NIOO-CEMO/DGW
C File: TEXAS.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,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 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 ###################################################
```

```
C######################################################################
C SENECA 1.85                                          (C) NIOO-CEMO/DGW
C File: DEPOSIT.FOR
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),pre(8),mic(8),sal(8)
       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


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

1        do 5 j=1,8
         if (dep(j).gt.1.e30)then
         dep(j)=xdiv(1,0.)
         else
         if (dep(j).lt.1.e-30)then
         dep(j)=xdiv(1,0.)
         else
         endif
         endif
5       continue


        do 10 j=1,8

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

        c(j)=poc(j)/(pm0*0.42*100*100*10)*100



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 *******
```

```
       teffdep(j)=1/exp(abs(temp(j)-31.5)/pm1)
       seffdep(j)=1/exp(abs(sal(j)-pm3A)/pm2)
       feffdep(j)=xdiv(xdiv(poc(j),dep(j)),xdiv(poc(j),dep(j))+pm4a)



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

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



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


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


          byodep(j)=30*
     &  (1-(exp(-pm7*dep(j))))
     &  *(pm61a*pred1(j)+pm62a*pred2(j)+pm63a*pred3(j))


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


10      continue


      END
C************************************************************************
```

```
C######################################################################
C SENECA 1.85                                           (C) NIOO-CEMO/DGW
C File: EPIGROW.FOR
C Date: 1-7-92
C######################################################################
      SUBROUTINE EPIGROW(TIME)
C
      IMPLICIT REAL(A-Z)
C Parameter:
      REAL TIME
C
C Submodel routine
C---------------------------------------------------------------------
C Declarations:

      INCLUDE 'TEXAS.DCS'
      INCLUDE 'EPIGROW.DCP'
      INCLUDE 'EPIGROW.DCV'
      INCLUDE 'XSIMO.DEX'
C
C---------------------------------------------------------------------
C Statements:
C
       integer j
       real epi(8),depi(8),mic(8)
       real temp(8),aleepi(8)
       real sal(8),depth(8)
       real p(8),n(8),si(8)
       real pred1(8),pred2(8),pred3(8)

       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(depi(1),depi1)
       equivalence(depi(2),depi2)
       equivalence(depi(3),depi3)
       equivalence(depi(4),depi4)
       equivalence(depi(5),depi5)
       equivalence(depi(6),depi6)
       equivalence(depi(7),depi7)
       equivalence(depi(8),depi8)

       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)

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)

p(1)=p1(time)
p(2)=p2(time)
p(3)=p3(time)
p(4)=p4(time)
p(5)=p5(time)
p(6)=p6(time)
p(7)=p7(time)
p(8)=p8(time)

n(1)=n1(time)
n(2)=n2(time)
n(3)=n3(time)
n(4)=n4(time)
n(5)=n5(time)
n(6)=n6(time)
n(7)=n7(time)
n(8)=n8(time)

si(1)=si1(time)
si(2)=si2(time)
si(3)=si3(time)
si(4)=si4(time)
si(5)=si5(time)
si(6)=si6(time)
si(7)=si7(time)
si(8)=si8(time)

depth(1)=depth1(time)
depth(2)=depth1(time)
depth(3)=depth3(time)
depth(4)=depth3(time)
depth(5)=depth5(time)
depth(6)=depth5(time)
depth(7)=depth7(time)
depth(8)=depth7(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  ****** cut unused parameter ranges  *****

         do 1 j=1,8
         if (epi(j).gt.1.e30)then
         epi(j)=xdiv(1,0.)
         else
         if (epi(j).lt.1.e-30)then
         epi(j)=xdiv(1,0.)
         else
         endif
         endif
1       continue



       do 5 j=1,8

c****     microfauna    ******

c production=max_production
c         *temp_limit
c         *light_limit
c         *nutrient_limit


c => nutrients limitation:
```

```
c atomic weight of P =30.9738 g/mol
c                  N =14.0067 g/mol
c                  Si=28.0855 g/mol
c                  C =12.011  g/mol

c n(j)     =u mol /l
c nlimi(j)=mg dw /m^2/10cm
c c:n:si:p=106:16:15:1
c ww=0.42dw=0.106Cw

      nlimi(j)=(n(j)/16)/(n(j)/16+micp1/16)
      plimi(j)=(p(j)/1)/(p(j)/1+micp1/1)
      silimi(j)=(si(j)/15)/(si(j)/15+micp1/15)

c   micp2 as a max primary production of the local area (mgC/h/m^2)



c promicv = g C   /m^2/day
c promic  = mg dw /m^2/month
c Stockwell 1989:pp.36, pp38:3-5 g C /day
c  so micp2 =>0.4 when 1 day=12 hours day time


      promicv(j)=micp2
c temperature limitation:
     &  *1/exp(abs(temp(j)-31.5)/micp3)
c light limitation (day length and ligh indensity):
     &      *dayl(time)
c nutrients limitation:
     &      *min(nlimi(j),plimi(j),silimi(j))


c transform the promicv become the food source for suspension
c    feeders (at 10 cm bottom water)
c    (mg dw /m^2/10cm/month)
c surpose that only suspesion 10 cm above sedimental surface can
c      be used by suspension feeders.

        promic(j)=promicv(j)*1000/0.42*30/10/depth(j)

 5     continue


c****  epigrowth feeder  ******


       do 10 j=1,8


c temperature and salinity effects  ****

       teffepi(j)=1/exp(abs(temp(j)-31.5)/pm1)
       seffepi(j)=1/exp(abs(sal(j)-pm3B)/pm2)
       feffepi(j)=xdiv(xdiv(promic(j),epi(j)),
```

```
     &                   xdiv(promic(j),epi(j))+pm4b)

        aleepi(j)=teffepi(j)*seffepi(j)*feffepi(j)

c production

        proepi(j)=epi(j)*pm8B/12*(1-epi(j)/pm9B)
     &           *aleepi(j)

         depi(j)=depi(j)+proepi(j)

          byoepi(j)=30*
     & (1-exp(-pm7*epi(j)))
     & *(pm61b*pred1(j)+pm62b*pred2(j)+pm63b*pred3(j))

         depi(j) = depi(j)-byoepi(j)

10    continue

      END
C*************************************************************************
```

```
C########################################################################
C SENECA 2.0                                             (C) NIOO-CEMO/DGW
C File: XSIM0.FOR
C Model: TEXAS
C Creation date: 18-1-1996
C This file contains the routine that calls all submodel routines (XSUBMODS)
C########################################################################
C
      SUBROUTINE XSUBMODS(TIME)
      REAL TIME
C
      CALL DEPOSIT(TIME)
      CALL EPIGROW(TIME)
      RETURN
      END
C of XSUBMODS
C
      INCLUDE 'XFORC.FOR'
      INCLUDE 'XBOUND.FOR'
      INCLUDE 'XWASTE.FOR'
      INCLUDE 'XTLAG.FOR'
```

```
C#######################################################################
C SENECA 2.0                                              (C) NIOO-CEMO/DGW
C File: XFORC.FOR
C Model: TEXAS
C Creation date: 14-6-1996
C This file contains all Forcing functions declarations.
C#######################################################################
C
      REAL FUNCTION PRED11(TIME)
      REAL TIME
      EXTERNAL XTIMSER
      PRED11 = XTIMSER(3,1,0,TIME)
      RETURN
      END
C of PRED11
C
      REAL FUNCTION PRED21(TIME)
      REAL TIME
      EXTERNAL XTIMSER
      PRED21 = XTIMSER(3,2,0,TIME)
      RETURN
      END
C of PRED21
C
      REAL FUNCTION PRED31(TIME)
      REAL TIME
      EXTERNAL XTIMSER
      PRED31 = XTIMSER(3,3,0,TIME)
      RETURN
      END
C of PRED31
C
      REAL FUNCTION PRED12(TIME)
      REAL TIME
      EXTERNAL XTIMSER
      PRED12 = XTIMSER(3,4,0,TIME)
      RETURN
      END
C of PRED12
C
      REAL FUNCTION PRED22(TIME)
      REAL TIME
      EXTERNAL XTIMSER
      PRED22 = XTIMSER(3,5,0,TIME)
      RETURN
      END
C of PRED22
C
      REAL FUNCTION PRED32(TIME)
      REAL TIME
      EXTERNAL XTIMSER
      PRED32 = XTIMSER(3,6,0,TIME)
      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
```

```
C######################################################################
C SENECA 2.0                                          (C) NIOO-CEMO/DGW
C File: XBOUND.FOR
C Model: TEXAS
C Creation date: 14-6-1996
C######################################################################
C Boundary conditions:
```

```
C######################################################################
C SENECA 2.0                                         (C) NIOO-CEMO/DGW
C File: XWASTE.FOR
C Model: TEXAS
C Creation date: 29-12-1995
C This file contains all Waste loads declarations.
C######################################################################
```

```
C#######################################################################
C SENECA 2.0                                          (C) NIOO-CEMO/DGW
C File: XTLAG.FOR
C Model: TEXAS
C Creation date: 3-6-1996
C This file contains all Time lag Functions declarations.
C#######################################################################
```

```
C######################################################################
C SENECA 1.85                                          (C) NIOO-CEMO/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)
      integer j
      equivalence (dep(1),dep1)
      equivalence (epi(1),epi1)
C
C----------------------------------------------------------------------
C Statements:
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

10    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***********************************************************************
```

```
C#######################################################################
C SENECA 1.85                                              (C) NIOO-CEMO/DGW
C File: DIATOM.FOR
C Date: 1-7-92
C#######################################################################
      SUBROUTINE DIATOM(TIME)
C

      IMPLICIT REAL(A-Z)

C Parameter:
      REAL TIME,temp(8),richr(8)
      real p(8),n(8),si(8), dmic(8),mic(8),depth(8),promic(8)
C
C Submodel routine
C----------------------------------------------------------------------
C Declarations:

      INCLUDE 'TEXAS.DCS'
      INCLUDE 'DIATOM.DCP'
      INCLUDE 'DIATOM.DCV'
      INCLUDE 'XSIMO.DEX'
C
C----------------------------------------------------------------------
C Statements:
C

        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(dmic(1),dmic1)
        equivalence(dmic(2),dmic2)
        equivalence(dmic(3),dmic3)
        equivalence(dmic(4),dmic4)
        equivalence(dmic(5),dmic5)
        equivalence(dmic(6),dmic6)
        equivalence(dmic(7),dmic7)
        equivalence(dmic(8),dmic8)

        p(1)=p1(time)
        p(2)=p2(time)
        p(3)=p3(time)

c         p(4)=p4(time)

        p(4)=p3(time)


        p(5)=p5(time)
```

```
      p(6)=p6(time)
      p(7)=p7(time)
      p(8)=p8(time)

      n(1)=n1(time)
      n(2)=n2(time)
      n(3)=n3(time)
c       n(4)=n4(time)
      n(4)=n(3)

      n(5)=n5(time)
      n(6)=n6(time)
      n(7)=n7(time)
      n(8)=n8(time)

      si(1)=si1(time)


c       si(2)=si2(time)

      si(2)=si1(time)


      si(3)=si3(time)
c       si(4)=si4(time)

      si(4)=si3(time)


      si(5)=si5(time)
      si(6)=si6(time)
      si(7)=si7(time)
      si(8)=si8(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)

      depth(1)=depth1(time)
      depth(2)=depth1(time)
      depth(3)=depth3(time)
      depth(4)=depth3(time)
      depth(5)=depth5(time)
      depth(6)=depth5(time)
      depth(7)=depth7(time)
      depth(8)=depth7(time)


        do 5 j=1,8
```

```
         if (mic(j).gt.1.e6)then
         mic(j)=xdiv(1,0.)
c         print *,'mic (',j,')  > 10^6'
         else
         endif

5       continue


        do 10 j=1,8

c=> microfauna

c => micb, main food source for epigrowth-feeding macroinfauna
c => the production of microfauna is calculated by chrolophy-a
c       and day lenth and temperature and nutrients

c production=max_production
c          *temp_limit
c          *light_limit
c          *nutrient_limit


c => nutrients limitation:
c atomic weight of P =30.9738 g/mol
c                  N =14.0067 g/mol
c                  Si=28.0855 g/mol
c                  C =12.011  g/mol

c n(j)    =u mol /l
c nlimi(j)=mg dw /m^2/10cm
c c:n:si:p=106:16:15:1
c ww=0.42dw=0.106Cw

       nlimi(j)=(n(j)/16)/(n(j)/16+micp1/16)
       plimi(j)=(p(j)/1)/(p(j)/1+micp1/1)
       silimi(j)=(si(j)/15)/(si(j)/15+micp1/15)

       promic(j)=mic(j)*micp2*30
c temperature effect:
     &          *exp((temp(j)-31)/micp5)
c light effect (day length and ligh indensity) effect:
     &         *dayl(time)*exp(-depth(j)/micp3)
c nutrients effect:
     &         *min(nlimi(j),plimi(j),silimi(j))

       promicv(j)=promic(j)*0.42/1000/30


c promic= gc /m^2/d

c       print *,'promic(',j,')  =',promic(j)


        richr(j)=xdiv(mic(j),
```

```
     &     min(n(j)*106/16/0.42/10*12.011*100*1000,
     &         p(j)*106/0.42/10*12.011*100*1000,
     &         si(j)*106/15/0.42/10*12.011*100*1000))

      resmic(j)=mic(j)*micp4*24*30
     &    *exp((temp(j)-31)/micp5)
     &    *(1+richr(j))**2

c      print *,'resmic(',j,')=',resmic(j)
c      print *,'sfodep(',j,')=',sfodep(j)


      dmic(j)=dmic(j)+promic(j)-resmic(j)
     &         -intepi(j)


10    continue

c      print *, 'diatom.for  done'

      END

C**********************************************************************
```

```
C######################################################################
C SENECA 2.0                                          (C) NIOO-CEMO/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,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,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,FRSHEAD,.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 consistancy
      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 maxiumum 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,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 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,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
      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 XRDCALPAR
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,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
      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
```

```
      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 permuation over NVASE and returns
C permuation 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) = XRAN(ISEED)
10    CONTINUE
C
C Index sort random array
C so that the index array will be a permution over NVASE
      CALL XSORTREALS(NVASE,RVASE,VASE)
C
      END
C End of XGETVASE
C
C
C*********************************************************************
      REAL FUNCTION XRAN(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)
      XRAN = FLOAT(ISEED)*RM
C
      END
C End of XRAN
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 IDMNCAL,IDMXCAL,MNCENTR,MXCENTR
      REAL CALNORM(XMXVASE)
      COMMON /XCBCAL/ IDMNCAL,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,XMXVAR
        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,XMXVAR
        VAR(I) = SAVVAR(I)
110   CONTINUE
C
C Generate new random values for random parameters
      CALL XRDCALPAR
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) then 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,IDMNCAL,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 ###################################################
```

```
C#######################################################################
C SENECA 2.0                                           (C) NIOO-CEMO/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 XSENS
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,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,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,FRSHEAD,.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 consistancy
      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,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
      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 influense 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,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
      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 file XMONTECARLO
C
C#### End of file ##################################################
```