PB92134667

Publication No. FHWA-RD-91-035

December 1990

U.S. Department
of Transportation

**Federal Highway
Administration**

Basic Study to Improve Speed and Efficiency of Vehicle/Barrier Simulations

Volumn I: Final Report

Office of Research and Development
Turner-Fairbank Highway Research Center
6300 Georgetown Pike
McLean, Virginia 22101-2296

Technical Report Documentation Page

1. Report No.
FHWA-RD-91-035

2.
PB92-134667

3. Recipient's Catalog No.

4. Title and Subtitle
BASIC STUDY TO IMPROVE SPEED AND EFFICIENCY OF VEHICLE/BARRIER SIMULATIONS, Volume I: Final Report

5. Report Date
December 1990

6. Performing Organization Code

7. Author(s)
R. L. Chiapetta

8. Performing Organization Report No.
4019FR

9. Performing Organization Name and Address
Chiapetta, Welch & Associates, Ltd.
9748 Roberts Road
Palos Hills, IL 60465

10. Work Unit No. (TRAIS)
3A5F4082

11. Contract or Grant No.
DTFH61-86-C-00025

12. Sponsoring Agency Name and Address
Concepts Research Design Division
Federal Highway Administration
Turner-Fairbank Highway Research Center
6300 Georgetown Pike; McLean, VA 22101

13. Type of Report and Period Covered
Final Report
June 1986 - December 1990

14. Sponsoring Agency Code

15. Supplementary Notes
Contracting Officer's Technical Representative (COTR) - Mort Oskard, HSR-20

16. Abstract

Some of the FHWA vehicle impact simulation computer programs were modified to improve their accuracy and/or efficiency.

The CRUNCH, GUARD and BARRIER VII programs as well as three versions of the NARD program were reviewed to identify modeling limitations and numerical procedures employed by each program. Other, commercially available, software was also reviewed. Based on the reviews, numerous recommendations were made to incorporate changes to the FHWA programs to increase their accuracy/efficiency.

A subset of the recommended improvements were implemented in three of the programs: GUARD 3.0, NARD 1.0 and PCNARD. This effort included development of a separate preprocessor program for GUARD and NARD. The modifications resulted in the attainment of various levels of success ranging from very little to a very significant increase in accuracy/efficiency, depending on the modification.

Other activities included a state-of-the-art review of soil/post interaction and development of general guidelines for the selection of integration time-step size.

This is volume I of a two-volume set. The other volume is FHWA-RD-91-036, Volume II: Appendixes.

17. Key Words
Roadside barriers, automotive vehicles, impact, computer simulations, accident, CRUNCH, GUARD, NARD, BARRIER VII

18. Distribution Statement
No restrictions. This document is available to the public through the National Technical Information Service, Springfield, Virginia 22161

19. Security Classif. (of this report)
Unclassified

20. Security Classif. (of this page)
Unclassified

21. No. of Pages
50

22. Price

Form DOT F 1700.7 (8-72)

Reproduction of completed page authorized

FOREWORD

The results of this study entitled "Basic Study to Improve Speed and Efficiency of Vehicle/Barrier Simulations" are documented in Final Reports FHWA-RD-91-035, "Basic Study to Improve Speed and Efficiency of Vehicle/Barrier Simulations," Volume I, Final Report and FHWA-RD-91-036, "Basic Study to Improve Speed and Efficiency of Vehicle/Barrier Simulations," Volume II, Appendixes. The work focused on deriving methods to enhance existing computer programs NARD, Crunch, and Guard. The resulting implementation of those methods is being documented in other reports. In view of the very limited interest in the study content, no distribution to the Regions or Divisions is planned.

Copies of the above reports will be made available from the National Technical Information Service, 5400 Port Royal Road, Springfield, Virginia 22161. A limited number of copies have been retained for internal office use.

RJ Betsold
R. J. Betsold
Director, Office of Safety and Traffic
Operations Research and Development

NOTICE

METRIC CONVERSION FACTORS

Approximate Conversions to Metric Measures

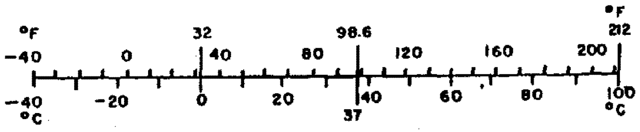
Symbol	When You Know	Multiply by	To Find	Symbol
		LENGTH		
in	inches	*2.5	centimeters	cm
ft	feet	30	centimeters	cm
yd	yards	0.9	meters	m
mi	miles	1.6	kilometers	km
		AREA		
in^2	square inches	6.5	square centimeters	cm^2
ft^2	square feet	0.09	square meters	m^2
yd^2	square yards	0.8	square meters	m^2
mi^2	square miles	2.6	square kilometers	km^2
	acres	0.4	hectares	ha
		MASS (weight)		
oz	ounces	28	grams	g
lb	pounds	0.45	kilograms	kg
	short tons (2000 lb)	0.9	tonnes	t
		VOLUME		
tsp	teaspoons	5	milliliters	ml
Tbsp	tablespoons	15	milliliters	ml
fl oz	fluid ounces	30	milliliters	ml
c	cups	0.24	liters	l
pt	pints	0.47	liters	l
qt	quarts	0.95	liters	l
gal	gallons	3.8	liters	l
ft^3	cubic feet	0.03	cubic meters	m^3
yd^3	cubic yards	0.76	cubic meters	m^3
		TEMPERATURE (exact)		
°F	Fahrenheit temperature	5/9 (after subtracting 32)	Celsius temperature	°C

---

*1 in = 2.54 (exactly). For other exact conversions and more detailed tables, see NBS Misc. Publ. 286, Units of Weights and Measures. Price $2.25, SD Catalog No. C13.10:286.

Approximate Conversions from Metric Measures

Symbol	When You Know	Multiply by	To Find	Symbo
		LENGTH		
mm	millimeters	0.04	inches	i
cm	centimeters	0.4	inches	i
m	meters	3.3	feet	ft
m	meters	1.1	yards	y
km	kilometers	0.6	miles	m
		AREA		
cm^2	square centimeters	0.16	square inches	in
m^2	square meters	1.2	square yards	yd
km^2	square kilometers	0.4	square miles	mi
ha	hectares (10,000 m^2)	2.5	acres	
		MASS (weight)		
g	grams	0.035	ounces	o
kg	kilograms	2.2	pounds	l
t	tonnes (1000 kg)	1.1	short tons	
		VOLUME		
ml	milliliters	0.03	fluid ounces	fl o
l	liters	2.1	pints	pt
l	liters	1.06	quarts	qt
l	liters	0.26	gallons	gal
m^3	cubic meters	35	cubic feet	ft^3
m^3	cubic meters	1.3	cubic yards	yd^3
		TEMPERATURE (exact)		
°C	Celsius temperature	9/5 (then add 32)	Fahrenheit temperature	



VOLUME I: FINAL REPORT

TABLE OF CONTENTS

TABLE OF CONTENTS (Continued)

LIST OF TABLES

VOLUME II: APPENDIXES
TABLE OF CONTENTS

APPENDIX 1: MILESTONE REPORT NO. 1 - REVIEW OF THE CRUNCH & GUARD PROGRAMS

VOLUME II: APPENDIXES
TABLE OF CONTENTS (Continued)

APPENDIX 1 (Continued)

APPENDIX 2: MILESTONE REPORT NO. 2 - REVIEW OF THE BARRIER VII PROGRAM

VOLUME II: APPENDIXES
TABLE OF CONTENTS (Continued)

APPENDIX 3: MILESTONE REPORT NO. 4 (Attachment 2) - SURVEY OF SOFTWARE/RECOMMENDATIONS

APPENDIX 4: MILESTONE REPORT NO. 4 (Attachment 1) - ASSESSMENT OF IMPROVEMENTS/RECOMMENDATIONS

VOLUME II: APPENDIXES
TABLE OF CONTENTS (Continued)

APPENDIX 6: TASK B.2.a REPORT - REVIEW OF SOIL/POST INTERACTION MODELS AND EXPERIMENTAL DATA

VOLUME II: APPENDIXES
TABLE OF CONTENTS (Continued)

APPENDIX 7: TASK B.2.b REPORT - RECOMMENDED SOIL/POST INTERACTION CONCEPT

APPENDIX 8: MILESTONE REPORT NO. 5 - DOUBLE PRECISION, RESTART AND NATURAL FREQUENCY MODIFICATIONS IN GUARD

VOLUME II: APPENDIXES
TABLE OF CONTENTS (Continued)

VOLUME II: APPENDIXES
TABLE OF CONTENTS (Continued)

APPENDIX 11 - LIST OF ILLUSTRATIONS

APPENDIX 12: MILESTONE REPORT NO. 7S1 - CORRECTION OF ERRORS AND SELECTION OF INTEGRATION TIME STEP IN GUARD

VOLUME II: APPENDIXES
TABLE OF CONTENTS (Continued)

APPENDIX 13: MILESTONE REPORT NO. 7S2 - NEW USER MANUAL AND BANDWIDTH MINIMIZATION PROGRAM DOCUMENTATION FOR GUARD

APPENDIX 14: MILESTONE REPORT NO. 8 - DOUBLE PRECISION, RESTART AND NATURAL FREQUENCY MODIFICATIONS IN NARD

APPENDIX 15: MILESTONE REPORT NO. 8S - EXTENSION OF RESTART FEATURE IN NARD

VOLUME II: APPENDIXES
TABLE OF CONTENTS (Continued)

APPENDIX 16: MILESTONE REPORT NO. 9 - ITERATION PROCEDURE AND VARIABLE TIME STEP IN NARD

APPENDIX 17: MILESTONE REPORT NO. 10 - BANDWIDTH MINIMIZATION AND SELECTION OF INTEGRATION TIME STEP FOR NARD

VOLUME II: APPENDIXES
TABLE OF CONTENTS (Continued)

APPENDIX 18: MILESTONE REPORT NO. 10S - EXPLICIT VERSUS IMPLICIT SOLUTION PROCEDURES

APPENDIX 19: MILESTONE REPORT NO. 11 - PCNARD SOIL/POST MODEL MODIFICATION

1. INTRODUCTION

The Federal Highway Administration (FHWA) has several computer programs that can be used to simulate crash events involving automotive vehicles and roadside barriers or obstacles.

The objectives of this contract were (1) to review a select number of these programs to identify areas where changes in assumptions, procedures or restrictions could lead to significant improvement in accuracy and/or efficiency of the simulations, (2) to make recommendations to modify the programs to achieve better performance, and (3) to implement a limited number of the recommendations.

The specific tasks conducted to meet these objectives consisted of the following:

(1) Reviewed GUARD 2.0, CWA/CRUNCH and BARRIER VII to determine what numerical procedures are employed, the inherent limitations due to modeling assumptions, and restrictions on model size.

(2) Surveyed available software, other than the FHWA simulation programs, to explore the possibility of adapting certain components of that software to the simulation programs to achieve improved performance from the viewpoint of accuracy and efficiency.

(3) Made an assessment of what the effect of potential program modifications (to GUARD, CRUNCH and BARRIER VII) would have on performance.

(4) Made recommendations on what changes should be made to GUARD, CRUNCH and BARRIER VII to improve accuracy and efficiency.

(5) Reviewed the NARD 1.0 (successor to CRUNCH) program vehicle/barrier interaction iteration procedure in preparation for incorporation of desired changes to the program.

(6) Implemented a select subset of recommendations made in item (4) in GUARD 3.0 and NARD 1.0.

(7) Developed a stand-alone preprocessing program for GUARD 3.0 and NARD 1.0 to minimize the bandwidth of the stiffness matrix of the barrier model.

(8) Performed a state-of-the-art review of soil/post interaction for highway barrier systems and made short- and long-range recommendations to improve the state of soil/post interaction modeling in the FHWA simulation programs.

(9) Replaced the faulty soil/post interaction model in PCNARD with a simpler, workable model.

The description of the work performed and results obtained on these tasks was reported in several interim reports in the form of Milestone Reports. They are provided as appendixes to this report. Each is self-contained, having its own numbering system for pages, figures, tables and equations, and its own set of references and, in some cases, contain appendixes (within the appendix).

The chapters (sections) of this report summarize the effort described in the appendixes.

2. PROGRAM REVIEW/ASSESSMENT/RECOMMENDATIONS

2.1 Review of Simulation Programs

2.1.1 Introduction

Three FHWA-sponsored crash simulation computer programs were reviewed. They are CRUNCH, GUARD AND BARRIER VII.(1,2,3) The review resulted in documentation of a general description of each code, identification of their modeling methodology and problem size limitations, and delineation of their salient numerical procedures. Detailed results of the review are presented in appendixes 1 and 2.

2.1.2 CRUNCH

The function of this program is to simulate collisions of both articulated and single-chassis vehicles with rigid and yielding barriers and roadside objects, accounting for the general three-dimensional response of the vehicles and barriers, and the large inelastic deformation of both vehicle and barrier. Various conventional and nonconventional barrier systems can be effectively modeled with the CRUNCH computer program. Vehicles which can be represented include passenger cars, buses, trucks and tractor-single and double semitrailers.

The vehicles are modeled with lumped masses connected by nonlinear spring and dashpot elements. The models are valid for three-dimensional response including large angular motion. The barriers are modeled using a three-dimensional finite element representation considering both nonlinear material and nonlinear geometric (large displacements) effects. Vehicle/barrier interaction forces are determined as a function of the overlap of elastic-plastic-frictional panels, representing the exterior vehicle surface, and of longitudinal contact lines on the barrier rails and posts.

Barrier modeling methodology limitations which were identified consisted of those in the following areas:

- Rail cross-section deformation.
- Blockouts.
- Tearing or fracture.
- Splice and bolt connections.
- Snagging.
- Crash cushions.
- Post-soil interaction characteristics.
- Direction of interaction forces.
- Material damping and strain rate effects.

Several problem size limitations of the program were revealed. For the vehicle module these included number of:

- Driver input points for steering and braking functions.
- Trajectory mode points.
- Road profile points.

For the barrier module, it was found to be limited in number of:

- Nodes.
- Elements (and related quantities).
- Displacement-specified nodes.
- Soil-post interaction nodes.
- Soil resistance function points.

In the interaction module, the maximum allowable number of vehicle contact panels is limited.

The temporal numerical integration procedures employed are mixed. For the vehicle, an explicit one-step Newton procedure is used. There are two basically different integration procedures supported for the barrier equations in CRUNCH, the Implicit and Explicit procedures. The version of the code based on the Implicit procedure, uses the Newmark-Beta method with constant time step. A variable, stiffness matrix update interval is allowed. A tangent stiffness matrix and a nonlinear force correction are used. The Gauss-Elimination technique is used for the matrix inversion process. The Explicit integration version of the code also uses a Newmark-Beta type of integration procedure with constant time step. The program has no restart capability, but does have a vehicle contact panel eligibility check.

2.1.3 GUARD

The GUARD program is similar to CRUNCH in many respects, but is generally not as comprehensive or refined, especially with respect to vehicle modeling and interaction force computation. As with CRUNCH, GUARD was designed to calculate the full three-dimensional response of both vehicle and barrier in a crash simulation. Most existing post-rail systems can be modeled; crash cushions, sign posts and luminaire supports cannot. Only nonarticulated vehicles can be considered.

The vehicles are represented by a single mass connected to the ground by nonlinear spring and dashpot elements to simulate the combination of suspension and tire deformability. The structure model used for post-rail systems is the same as that in CRUNCH, i.e., a three-dimensional finite element model considering nonlinear material and geometric effects. The vehicle/barrier interaction force model is similar to that in CRUNCH in that it employs massless contact panels distributed over the contact side of the vehicle which interact with a number of contact lines scribed on the surface of the barrier rail. However, the interaction force calculation strategy is much less sophisticated than that in CRUNCH. On the other hand, GUARD features explicit representation of car bumpers whereas CRUNCH does not.

Two configurations of the GUARD program are currently available, the first employing both the vehicle/interaction and the barrier modules for the full simulation capability and a second, intended for use with rigid longitudinal barriers, employing an alternate main program and omitting the barrier module entirely.

Problem size limitations of GUARD include number of:

- Nodes.
- Elements (and related quantities).
- Soil resistance functions.
- Road profile points.
- Vehicle contact panels.

The time-wise numerical integration procedure for the vehicle model is the same explicit, simple one-step Newton method as found in CRUNCH. The solution of the barrier equations is based on an implicit formulation and contains the same major features as cited for the CRUNCH code. GUARD was found to have a complete restart feature, but had no vehicle contact panel eligibility check. The integration procedures for both vehicle and barrier are limited to constant time steps, with the time steps for vehicle and barrier required to be the same.

2.1.4 BARRIER VII

This program was the first one, developed for the FHWA for the simulation of crash events, which was based on a finite element representation of the barrier model. The program was designed to predict the in-plane, two-dimensional response of both vehicle and barrier. It employs a nonlinear finite element procedure for the analysis of the barrier structures and a simple, three-degree of freedom dynamic model for the impacting vehicle. The barrier structure is represented by an assemblage of beams, cables, posts, springs, links and damping devices, and impact between the vehicle and barrier is achieved by means of deformable boundary springs on the vehicle and preselected contact zones on the barrier.

The vehicle is modeled as a single mass with in-plane rotational inertia. There is no representation of the suspension. The vehicle is allowed to interact with the barrier by means of a specified number of contact points on the car boundary. A discrete nonlinear spring is associated with each point. The barrier is idealized as a framework of arbitrary shape, lying in the horizontal plane. The framework can be composed of the different types of members mentioned in the preceding paragraph. These members may be superimposed in a variety of ways to represent some unconventional as well as conventional barrier systems.

The primary modeling methodology limitations of the program are that it is restricted to two-dimensional, in-plane, response calculations and that the vehicle model is extremely simplistic. In addition, it was found to be deficient in the following areas:

- Tire skid force algorithm.
- Steerable front wheels.
- Articulated vehicle.
- Bumper model.
- Crash cushions.
- Hysteretic effects in barrier.
- Material strain hardening effect.
- Moment-curvature, thrust-extension relations.
- Large deflection capability.
- Convergence and stability of solution procedure.

The problem size limitations consist mostly of restrictions on the number of barrier nodes and elements.

The numerical integration procedure for the vehicle is an explicit formulation based on the constant acceleration method. The numerical solution procedure for the barrier relies on an implicit formulation employing an incremental, iterative strategy. The integration may be accomplished with either a constant or mid-point constant acceleration method with a variable time step.

2.2 Survey of Software/Recommendations

2.2.1 Introduction

A survey was conducted to <u>identify</u> other available computer software that might be used to replace major <u>modules</u> or <u>submodules</u> of CRUNCH, GUARD and BARRIER VII to increase their accuracy and/or efficiency. In addition, a review was made of newly available computer <u>hardware</u> which can be used to increase the speed of computation of these programs. Finally, an available pre- and post-processor was identified which could be used to decrease engineering labor time associated with crash simulation and to decrease mainframe CPU costs.

The criteria used in identifying the "other" available software was that it (1) perform essentially the same function that is currently being performed by the comparable software in BARRIER VII, GUARD or CRUNCH; (2) has the potential for increasing the accuracy and/or efficiency (speed) of the programs; and (3) can be adapted to one or more of the three programs under consideration without a <u>major</u> effort involved.

Recommendations regarding possible usage of the other software are made throughout the discussion.

A detailed description of the survey is provided in appendix 3.

The following summary of the survey is divided into four parts:

(1) Modules.
(2) Sub-Modules.
(3) New Hardware Architecture.
(4) Pre- and Post-Processing.

2.2.2 Modules

There are three logical divisions (vehicle, interaction, barrier) inherent in the vehicle-barrier simulation programs. All three programs (CRUNCH, GUARD, BARRIER VII) recognize these distinctions and to some extent reflect it in the organization of the programs. the CRUNCH, in particular, was designed from the outset with three relatively independent modules in mind for these three aspects of the problem. In view of the continuing, massive development efforts on computer-based analysis and simulation throughout the scientific community, this modularity raises the possibility of the "swapping" of modules at this level.

Some possible sources of large-scale modules from the commercial sector are as follows:

<u>Structural</u>

Commercial, general purpose finite element codes such as ANSYS, MSC/NASTRAN, ABAQUS, MARC, ADINA (refs. 4 through 8).

Public or closely held nonlinear finite element codes such as WHAM, WRECKER, DYCAST, NONSAP, COSMIC/NASTRAN (refs. 9 through 13).

All of these codes have the desired capabilities which the structural modules of CRUNCH, GUARD and BARRIER VII possess, namely, they can accommodate large deflection, three-dimensional, nonlinear response of a structural system modeled with beam elements. However, they are much more general because they have a more comprehensive element library including plate, solid and special elements.

Vehicle Dynamics

Four categories of available vehicle dynamics programs were identified. Some programs in each category were reviewed; they are:

1. General Purpose Rigid Body Vehicle Dynamic Codes
    - DRAM-ADAMS
    - DADS
    - Army (TACON Labs) - in house
    - NASA - in house
2. Vehicle Crash Simulation Codes
    - One-Dimensional Lumped Mass Models
        - (a) NHTSA Model
        - (b) Battelle
        - (c) Mini-Cars
        - (d) Calspan
        - (e) Dynamic Science
    - Framework Models
        - (a) Calspan
        - (b) HSRI (McIvor)
        - (c) CRASH (Philco-Ford)
    - Finite Element Models
        - (a) WRECKER (NHTSA)
3. Vehicle Handling Codes
    - HVOSM (Calspan)
    - AVDS/TDVS (IITRI)
    - PHASE4 (HSRI)
    - Hybrid-Computer Handling Program (APL)
4. Accident Reconstruction Codes
    - SMAC/CHISMAC
    - CRASH3 (Calspan/NHTSA)

Interaction

No stand-alone, independent modules for the interaction process (as a distinct computational phase) were identified. The general subject of interaction between distinct models and parts of models has received increasing attention and a variety of techniques are contained in the literature and in specialty analyses for fluid-structure, soil-structure and other interaction-type problems.

While the concept of module swapping at one of the three logical division levels (vehicle, interaction, barrier) appears, at first sight, to be attractive, one must be aware of the following points:

- General purpose analysis packages, both structural and rigid body dynamics, carry large computational overhead because of their generality and, in addition, have strict and closed data formats making large-scale integration with other modules difficult.

- There is never complete independence of vehicle and barrier modules because any interaction strategy necessitates assignment of certain interaction attributes (e.g., interaction lines, body springs, crushable envelopes, crush panels, etc.) to the barriers and others to the vehicles. Another major ingredient of this strategy is an algorithm for calculation of interaction force magnitude and direction as a function of the response of the interaction attributes of the barrier and vehicle. Hence, current interaction strategies are intimately connected with specific vehicle and barrier models.

- Within the suite of FHWA crash simulation programs, only CRUNCH is reasonably modular; vehicle and interaction functions in GUARD and BARRIER VII are considerably entwined.

2.2.3 Sub-Modules

There are many mathematical models in CRUNCH, GUARD and BARRIER VII simulating a physical phenomenon within a given module (vehicle, barrier or interaction); these routines might be called sub-modules. Several available routines that could be considered for possible inclusion in one of the three programs of interest to improve their accuracy were addressed.

Most of the items considered concern simulation of a very complex physical process which is but only one component of the overall physical event modeled by the given program module. these "sub-events" are, for the most part, currently modeled on a semi-empirical basis as opposed to using a model based on first principles. In the following, we identify some of these areas and discuss the pros and cons of alternatively replacing the semi-empirical model with a first principles model.

Within the structural (barrier) module of these programs there are two notable sub-modules which warrant attention: (1) the soil-post interaction force model, and (2) the models handling crushable components (e.g., crushable, energy-absorbing rings). Although there are first principles techniques available for these models, they are not recommended for inclusion in the programs because they are either too inordinately elaborate and/or their accuracy has not been established. Rather, it is suggested that, at this point in time, they be used to supplement test data to derive semi-empirical models. There are also other areas within the structural model where these comments may also apply, e.g., models representing splices in longitudinal rail and slip or frangible bases in poles.

In the vehicle module, a similar situation exists with respect to the tire-roadway interaction force model, where the "roadway" can be interpreted to be the road, a curb, or the surface of a concrete-shaped barrier. Also within the vehicle module, there have been attempts to simulate the effect of swinging or sloshing payloads. They have enjoyed only limited success, so it is not clear that these actions should be represented by the models so employed.

For the vehicle-barrier interaction module, a literature search revealed alternate methods for the computation of interaction forces at the interface between solid media. However, these were primarily applicable to situations where both bodies in contact were represented by finite element models, which is not the case for CRUNCH, GUARD and BARRIER VII.

2.2.4 New Hardware Architecture

A brief study was conducted to determine the potential benefit of using relatively new hardware architecture in connection with the crash simulation programs of interest. In particular, the possible use of vector processors employing a combination of so-called parallel and pipeline modes of computation was explored.

The relative performance of these programs on vector processors compared to a scalar processor is dependent on several factors, including (1) the percent of the program which can be vectorized and (2) the length of the vectors.

Early studies on vectorizing showed that significant gains in computation time were possible for operations on global arrays,but, for nonlinear finite element programs, major reorganization of program structure and numerical techniques may be necessary to realize the full potential of vector processors.

2.2.5 Pre- and Post-Processing

A few commercial and private-sector software packages which would be suitable for pre- and post-processing activities for the CRUNCH, GUARD and BARRIER VII simulation programs were reviewed. They could be used to either supplement or supplant the packages currently being used to perform pre- and post-processing on either PC or mainframe computers. One of the software packages reviewed is especially notable because it is much simpler to use than the GRAFIX and PREP programs which are currently used for post-processing for CRUNCH and GUARD.(14,15,16)

2.3 Assessment of Expected Improvement/Recommendations

2.3.1 Introduction

An assessment was developed of the improvement in run cost and accuracy that can be attained if the CRUNCH, GUARD and BARRIER VII programs are expanded in problem size and improved integration subroutines are inserted, and/or government or commercially available software is acquired, made operational at Johns Hopkins Applied Physics Laboratory (APL) and the impact problem re-expressed to fit these revised programs. The assessment emphasized accuracy and run-time improvements as a function of problem size and numerical improvements.

Based on this assessment, general long-range recommendations were outlined for action to improve the three computer programs as well as specific recommendations to be implemented on a subsequent task on this contract. In addition, criteria were developed to provide a measure of the improvement achieved due to changes that would be made to the programs.

In the following, a summary is given of the assessment process, the recommendations made, and the evaluation criteria developed. Details of this effort are given in appendixes 4 and 5.

2.3.2 Assessment

The assessment of the degree of improvement that might be achieved was based on the following sources of information:

- Prior knowledge of the codes.
- Review of the codes.
- Timing experiments.
- Implicit vs. explicit solution strategy study.

Two series of computer runs were made during the assessment. The first series of runs consisted of timing experiments on all three codes (CRUNCH, GUARD and BARRIER VII). Changes were made to the programs to determine exactly how much CPU time is spent in several major subroutines in each program during a typical simulation. Numerical results were then generated for one sample problem for each program using input data which was available in FHWA's Roadside Safety Library. The information from this series of runs was useful to pinpoint the areas of the three programs which use the greatest proportion of CPU time, suggesting that effort be concentrated on these areas to achieve the biggest payoff in program efficiency. The second series of computer runs were made during a study of Implicit vs. Explicit solution methods using only the CRUNCH program (since GUARD and BARRIER VII do not offer an optional Explicit solution procedure). The effect of time step was also included in the study.

The results of the two series of computer runs provided input to the process of assessing potential improvements in the three computer programs of interest here. Improvements in five general areas were considered:

(1) Modeling Methodology.
(2) Problem Size Capacity.
(3) Numerical Methods.
(4) Other Commercially Available Software.
(5) New Hardware Architecture.

As specified in the contract, emphasis was placed on only the Problem Size and Numerical Method categories.

The criteria used for considering a potential improvement in any of the five areas was that the item considered have potential for improving either the efficiency or the accuracy (perhaps at the expense of reduced efficiency) of one or more of the three computer programs of interest.

2.3.3 Recommendations

The following is a list of areas for which it was recommended that action be taken with respect to each of the five general areas to improve the efficiency or accuracy of the three computer programs. These are to be considered as general recommendations and not confined to those to be implemented on later phases of this contract.

The title of each item in the list is followed by letters enclosed in parentheses. These letters indicate which program(s) the item applies to, where:

- **C = CRUNCH.**
- **G = GUARD.**
- **B = BARRIER VII.**

<u>Modeling Methodology (Accuracy)</u>

M1. Blockouts (C,G)
M2. Tearing/Fracture (C,G)
M3. Snagging (C)
M4. Crash Cushions/Sand Troughs (C)

M5. Post/Soil Interaction Characteristics (C,G,B)
M6. Material Characterization (C,G,B,)
M7. Vehicle/Barrier Interaction Forces (C,G,B)
M8. Tire/Concrete-Barrier Interaction Forces (C,G)
M9. Vehicle Suspension Model (C)
M10. Vehicle Contact Panels (C,G)

Problem Size (Accuracy)

P1. General/Rearrangement (C,G,B)
P2. Storage Requirement Algorithm (C,G,B)
P3. Road Profile (C,G)
* P4. Nodes and Elements (C,G,B)
* P5. Integration Points (C,G)
* P6. Contact Lines/Points (C,G)
P7. Soil Property Sets (G,B)
* P8. Vehicle Contact Surfaces/Points (C,G)
P9. Boundary Nodes (C,G,B)
P10. Post-Soil Nodes (C,G,B)
P11. Eligible Elements (C)
P12. Soil Property Curve (C,G,B)

Numerical Procedures (Efficiency)

N1. Double Precision (C,G)
N2. Restart (C)
N3. Frequency Calculation (C,G,B)
* N4. Variable Time Step (C,G)
N5 Bandwidth Optimizer (C,G,B)
N6. Dynamic Storage Allocation (C,G,B)
* N7. Linear Equation Solver (C,G,B)
N8. Underflow/Divide Check Errors (C,G)
N9. Implicit vs. Explicit (C)
N10. Time Step Magnitude (C,G,B)
N11. Eligible Contact Elements (C,G)
* N12. Contact Line/Panel Interaction (C)
* N13. Formation, Triangularization and Solution of Stiffness Matrix (B)

Vectorization (Efficiency)

* V1. CRUNCH and GUARD
* V2. BARRIER VII

The items marked with an asterisk (*) are those which were recommended for implementation on the current contract.

2.3.4 Evaluation Criteria

In order to evaluate the amount of "improvement" to the codes produced by changes made to them on this contract, certain evaluation criteria had to be established. They were as follows.

To evaluate the difference made by "accuracy-type" program changes, it was recommended that the following quantities be compared for before vs. after simulations and, where applicable, for test results:

Vehicle-Related Quantities

- Trajectory y vs. x.
- Heading (or yaw) angle history.
- Roll angle history (for rigid barriers only).
- Exit angle (for comparison with test data only).
- Exit speed (for comparison with test data only).

Barrier-Related Quantities
(for deformable barriers only)

- Horizontal plane deflection profiles at selected times.
- Maximum deflection.

For non-time history comparisons, such as maximum barrier deflection, a comparison of magnitudes of the quantities is to be made. Comparison of time histories of a given quantity may be made by comparing magnitudes of characteristic parameters, such as maximum value and "rise" time to maximum, or by using a statistical measure of the closeness of the curves.

Where possible, CPU simulation times for before vs. after should be compared so that one can determine what the increase in accuracy cost in terms of increase in running time.

To evaluate the difference made by "efficiency-type" program changes, it was recommended that the before vs. after simulation CPU running times be compared.

3. REVIEW OF SOIL/POST INTERACTION MODELS AND EXPERIMENTAL DATA

3.1 Introduction

A review was made of selected literature on analytical models and experimental data related to the behavior of guardrail posts mounted in soil. This review was undertaken with the purpose of identifying concepts and techniques for use in an improved treatment of soil/post behavior in the CRUNCH/NARD family of computer programs.(1,17,18)

The subject of soil/post interaction for guardrail systems is part of the much broader fields of soil/structure interaction and soil foundation mechanics. In view of the enormous literature of these fields, and of those aspects of soil/post behavior that are specific to guardrail applications, this review was limited to sources related fairly specifically to this application. It will be noted, however, that several of the research reports reviewed contain extensive surveys of the broader literature and assessments of its relevance to guardrail posts.

It will be noted, also, that this review made a somewhat arbitrary distinction between analytical and experimental investigations. Several of the research studies cited as being analytical in nature contain more or less extensive experimental results used as a comparison basis for the analysis. In addition, many of the sources dealing with the design, development and testing of guardrail systems are inherently experimental and semi-empirical in nature, due to the complexity of the overall problem, and were cited as such. It will also be noted that large-scale simulation computer programs which contain analytical representations of soil/post interaction as part of a broader analysis of vehicle/barrier impact are discussed here in a separate category.

Within this scope, the review presented pertinent results from what are broadly termed experimental sources; a review of certain research that is of primarily analytical interest; a summary of the manner in which soil/post interaction is implemented in the major vehicle/barrier simulation programs currently available and, finally; a summary of results relevant to the CRUNCH/NARD family of programs.

A summary of each of these areas is given in the next four sections, followed by an abbreviated statement of recommendations. Detailed results of the review topics and the full recommendations are provided in appendixes 6 and 7.

3.2 Experimental Data

Current guardrail systems, and the associated guides and standards are based largely on extensive series of full-scale crash tests and ad hoc experimental studies. This data, specifically directed toward the soil/post interaction problem, occurs sporadically throughout these studies and for the most part has been obtained in the form of post overturning load-deflection data for strong post systems. Such data is largely site specific and correlated to soil conditions only in terms of the AASHTO "strong/weak" categories (with one exception). Results for other post systems (i.e., weak posts) is for the most part obscured by test procedures which intermingle post-bending behavior with soil/post interaction.

3.3 Analytical Approaches

Several studies have been reviewed which represent three alternate analytical approaches to post/soil interaction. These three approaches represent the points of view of traditional soil-mechanics, visco-elastic/plastic geological models, and large-scale, fine resolution computational procedures (finite difference, finite element, etc.). Of these, the soil-mechanics approach embodied in Ref. 19 by TTI, provides reasonable correlation with post overturning results within the context of conventional soil mechanics.

In addition, several "elementary" representations of soil/post interaction were considered largely to clarify approximate computational models currently used in large-scale crash simulation programs.

3.4 Implementation in Crash Simulations

Also, a survey was made of techniques and computational models used to represent soil/post interaction effects in various barrier crash simulation programs. These implementations typically consist of a nonlinear generalization of the elementary analytical treatments. Primary attention is given to the six degree of freedom, uncoupled model employed in the CRUNCH program and, in considerable detail, to the attempted generalization of this approach to a coupled, subgrade modulus formulation reported in connection with NARD. Although the latter model would appear to be a modest and reasonable improvement of previous models, considerable difficulty has been reported with its implementation and extensive comment and criticism were offered in this connection.

During the review, several errors in the NARD soil model were revealed. They occur with respect to both the interpretation of the model and the implementation of it. The reported difficulties in the use of this model is largely attributable to these errors.

3.5 Prospects for CRUNCH/NARD Development

Based on the results of this review, it is possible to identify a spectrum of approaches available for the representation of soil/post interaction effects in the CRUNCH/NARD barrier crash simulation program. This spectrum ranges from very approximate, but simple, models to very elaborate, but computationally demanding techniques, and strong arguments can be made on behalf of various approaches. The differences among approaches are seldom a question of right and wrong, but more a question of available resources, the state of the art, underlying purpose of the program, etc. With this observation in mind, the following broad comments were offered as preamble to the submission of alternate approaches to the treatment of soil/post interaction:

* **Perspective and Emphasis**
The CRUNCH/NARD program, taken as a whole, is a collection of complex, highly nonlinear computational techniques. Major components in this simulation are a nonlinear dynamic vehicle simulation: nonlinear algorithms for vehicle/structure interaction and a dynamic, structural analysis of deformable barriers that is nonlinear in both material and geometry. Soil/post interaction effects are one nonlinear element within the structural module and its development should not particularly out-pace the development of other aspects of the program to the possible detriment of the whole.

* **Balanced Representations**
Certain aspects of the simulation require a representation of extremely complex and nonlinear phenomena which cannot, at present, be represented in a very satisfying manner by an approach from first principles. Vehicle/barrier interaction is one such area and soil/post interaction may be another. The response of strong post systems inherently involves severe deformations in the soil, for instance, and the use of even the most convincing linear soil/structure interaction technique is unlikely to provide acceptable results. That is, there is a constant tension in the development of this program, between the desire for an analysis from first principles and the expedient use of modeling artifices to retain "real" effects in the program.

* **Program Development**
On the other hand, there is no reason why the CRUNCH/NARD program should not be subject to continuous research and development, with efforts focused on various aspects of the program. It might be feasible to maintain two "official" versions of the program: a released version for general use and a research version in which specific improvements could be tested without jeopardizing the overall usefulness of the general release version.

Several alternate approaches to the treatment of soil/post interaction in the CRUNCH/NARD barrier crash simulations programs were presented. As noted above, these approaches vary considerably in terms of complexity, research effort and level of approximation.

(1) CRUNCH Model

The simple, six degree of freedom model currently installed in CRUNCH is capable of representing soil/post interaction and other post base effects, but is basically a means of "replaying" test data. It is far removed from a first principles approach and depends in part on the skill and modeling strategy of the user.

(2) NARD Model

The current NARD program contains a relatively simple single node embedded post subgrade modulus model, which, if properly implemented, could provide a modest improvement over the CRUNCH model while retaining the ability to represent fully nonlinear soil/post properties. It appears, however, that this model is not usable as implemented because of existing errors and requires further study, reformulation and testing.

(3) Nonlinear Subgrade Element

It would seem feasible to develop a fully nonlinear subgrade beam element capable of providing a nonlinear soil mechanics-based interaction model similar to that reported in reference 19. This model would consist of the full nonlinear beam element currently employed in CRUNCH/NARD with the addition of nonlinear foundation properties derived from existing treatments based on a soil mechanics or other point of view. The full treatment of an embedded post would require several subgrade elements to adequately represent each post. A substantial research effort would be required to formulate, implement and thoroughly validate such a model.

(4) Continuum Analyses

We have noted at several points in this review, that there always remains the possibility of approaching the soil/post interaction problem by means of three-dimensional, large-scale, finite element based numerical procedures. To this point, no successful attempts have been reported, but the constant improvement in finite element technology and computers may eventually provide such a possibility. it would appear, however, that this approach would focus on a single, isolated post configuration and would serve as an alternate to post testing as a source of data, rather than as a means to represent the soil around each post in a complex crash model.

(5) Rate Effects

Although we have mentioned the possible importance of rate effects in soil/post interaction at several points in this review, we have not presented any specific implementation of this, although the development of a full subgrade interaction element would necessarily contain such effects. We note, however, that the CRUNCH/NARD programs contain no provisions for a consistent treatment of rate effects in the barrier module and that this greatly inhibits any attempt to introduce such effects within the element formulation. A reformulation of the basic finite element solution algorithm would be required to provide rate effects throughout the barrier module.

3.6 Recommendations for Post/Soil Model Development

This section outlines recommendations for further development and refinement of the post/soil interaction models for implementation in the CRUNCH/NARD vehicle/barrier crash simulations. We distinguish here between recommendations for development beyond the scope of the present project activities (i.e., long-term recommendations) and those for action to be undertaken in the current project.

3.6.1 Long-Term Recommendations

Recommendations which, in the present context, are considered long-term are:

(a) Acquire additional, fundamental data relating to soil/post interaction. Such data should be obtained in a systematic and controlled manner with the specific purpose of supporting the further development and evaluation of analytical representations. Further, the experimental programs should include ranges of soil conditions, post types and load conditions which are typical of actual barrier construction and installation conditions.

(b) Develop a fully nonlinear subgrade beam element along the lines described in section 3.5(3) for incorporation in CRUNCH/NARD as an alternate analysis option. This element might employ the nonlinear beam element currently available in the programs, in conjunction with a nonlinear soil interaction model similar to that reported in reference 19, or an equivalent interaction formulation.

(c) Explore the possibility of using large-scale, three-dimensional finite element techniques for the high resolution analysis of selected post/soil interaction problems as an additional source of data and as a means of gaining additional insight into this problem area.

The following notes are offered in connection with the long-term recommendations.

*** Items (a), (b) and (c) above represent a long-range plan to continually incorporate more and more first principles into the soil/post interaction model. The order of progression suggested is to first start with Item (a) and then move on to Item (b) and eventually to Item (c). The submittal of this plan is predicted on the assumption that soil/post interaction will remain a significant factor in the performance of existing barrier systems. For example, it has been shown that performance of strong-post barrier systems is quite sensitive to soil/post interaction characteristics but not so in the case of weak-post or concrete barrier systems. Therefore, as long as strong-post systems continue to represent a significant percentage of the existing population of barriers, it is justified to strive to implement a long-range soil/post model improvement program.

*** The level of detail of the soil/post interaction model should be consistent with levels in other models used to represent equally significant physical phenomena (e.g., vehicle/barrier interaction) in the program.

*** Both items (b) and (c) above require considerable amount of research and validation before becoming viable candidates for soil/post models to be incorporated in FHWA's crash simulation programs. It is advisable to retain a working soil/post model in the program until such time that an experimental version of a model is developed to satisfaction. In this way the FHWA will always have at least some capability in this area and will allow for ever increasing capability as the developing technology permits.

*** The development of class b or c type models should include a review of soil/structure interaction research conducted in connection with other application areas such as pile driving, earthquake excitation of embedded structures and response of missile silos to air blast or ground shock induced by high explosives or nuclear weapons. Years of research have already been performed in each of these areas. The review would serve to

(a) determine the state of the art of the extent of success or failure in modeling soil/structure interaction phenomena, and (b) identify those computational techniques which have proved to be the most successful in analyzing these phenomena.

3.6.2 Short-Term Recommendations

In terms of activities on the present project, we strongly recommended the immediate replacement of the post/soil model currently implemented in the NARD programs. Our previous review of these programs clearly indicated that the piece-wise linear subgrade model employed is unusable as it stands. Furthermore, the existing implementation is so badly executed, both in detail and in its fundamentals, that it probably could not be salvaged within the constraints of the present project. In order that the FHWA has available a working program with some representation of a post/soil interaction, we proposed to replace the current attempt at a subgrade modulus model, in its entirety, with the six degree of freedom, uncoupled model as originally installed in the CRUNCH program.

4. GUARD PROGRAM CHANGES AND DOCUMENTATION

4.1 Introduction

In section 2.3.3 a list of recommendations was presented for changes to be made to the GUARD program. The FHWA selected the following areas to be addressed:

- Double precision.
- Restart capability.
- Natural frequencies of barrier model.
- Variable time step.
- Vehicle panel impact eligibility.
- Guidance in time step selection.

In the following sections, the work accomplished in each of these areas is summarized. This is followed by a section describing some errors detected in the program, a section outlining the new program input required as a result of the program changes made, and finally by a section commenting on the adaptation of the program changes to rigid barrier simulations. Details of all work performed on the GUARD program may be found in appendixes 8 through 13.

4.2 Double-Precision

A sample crash simulation problem was run with GUARD using double-precision in the calculations. Double-precisioning was accomplished in two different ways: (1) making detailed changes to the FORTRAN code, and (2) letting the system invoke double-precisioning by means of selection of appropriate parameters in the IBM JCL.

Results showed very little difference in vehicle response (especially during the vehicle crash phase) predicted by the two double-precision methods and only small differences between single and double-precision results. it was also found that the difference in run cost between single and double precision runs on the IBM mainframe was approximately the same or, at most, only slightly more for a double precision run.

It was recommended that double-precisioning be employed for GUARD simulations, and that it be accomplished on an IBM mainframe using a JCL parameter rather than making changes to the FORTRAN code.

4.3 Restart Capability

The GUARD program had an existing restart feature in it, i.e., a simulation could be restarted at any one of a number of pre-specified checkpoints (time steps) after a simulation had been terminated. This feature was already incorporated in the code in the version of the code supplied to the FHWA in reference 2. On a subsequent contract, changes were made to GUARD to force it to generate output files that could be used to perform post-processing of results with the GRAFIX and PREP programs.(15,16)

On the present contract, the operational status of the restart feature was checked. It was found that the restart feature per se worked properly but it was discovered that the output files for GRAFIX and PREP are not properly generated when a simulation has been "restarted," i.e., as it stands, the GRAFIX and PREP programs cannot be used to perform post-processing functions from a GUARD simulation which has been "restarted" at least once.

In view of the above-described problem with the graphics files, the program was modified to allow the user the option of turning off the graphics portion of GUARD. This was accomplished by requiring the input of a flag, called IGRAPH. If IGRAPH=0, the graphics features of GUARD are ignored (i.e., the generation of output files for GRAFIX and PREP). The value of IGRAPH is read in subroutine READIN as the last "card" (line) in the standard GUARD input data. To incorporate this feature, changes had to be made to only three subroutines; READIN, SOLVE and OUTPUT.

It should be noted that there are standard output files (not those for GRAFIX and PREP) in GUARD and that these <u>are</u> properly positioned for one or more restarts, so that any post-processing package that can read the format of these files can successfully make plots of a restarted simulation, no matter how many restarts were involved.

4.4 Natural Frequencies

Changes were made to the program to calculate elastic "nodal frequencies" of the <u>barrier</u> model. The highest nodal frequency is an approximation to the highest frequency of the barrier model and governs selection of the largest time step possible to avoid numerical instability difficulties.

A routine was added to the program to calculate the periods associated with all six degrees of freedom for each node in the barrier model. The smallest period is printed out along with the corresponding node number and degree-of-freedom. This calculation is performed before the integration phase of the program, but after the assembled stiffness matrix has been adjusted for boundary conditions.

To make use of this calculation, the user should make a "pre-integration" run with the program, i.e., set MXSTEP=0 (no integration steps). In this run, it is suggested that the time step be set to a large value (e.g., 1000 sec). This makes the calculation of periods more accurate.

It is recommended that a constant time step of integration be selected that is no larger than the period associated with the highest nodal frequency of the barrier system.

The program changes to GUARD specifically consisted of:

(1) Adding a subroutine named FREQ.
(2) Calling FREQ from Subroutine SOLVE, immediately after the CALL BOUND statement.

4.5 Variable Time Step

A variable time step (VTS) feature was added to GUARD. The specific time step selection procedure implemented represents an attempt to maintain uniform accuracy of computation from time step to time step while at the same time preventing numerical instability from occurring.

The VTS feature is restricted to flexible barrier impact (as opposed to "rigid" barrier impact) and may be turned on optionally by the program user. When selected, the VTS option requires the input of a new parameter, ETSS. An approximate formula was developed to allow the user to compute the parameter as a function of desired solution accuracy, or vice versa.

A sample simulation problem was solved to verify the VTS feature. Using the above-mentioned formula, a solution error was predicted which turned out to be within 3 percent of "actual." Furthermore, in this sample problem, the use of the VTS feature eliminated the numerical instability that occurred in the comparable constant time step run of the same problem.

The subroutines in GUARD which were affected by incorporation of the VTS feature are SOLVE and READIN. The subroutine READIN was changed to require the input of three new parameters, they are:

- ITMSTP - This is the VTS flag to turn the feature on or off.

- ETSS - This is the parameter used to calculate a new time step. It is related to solution accuracy.

- TIMEMX - This is the maximum allowable solution time in the current simulation. Note, this is completely unrelated to the VTS feature.

The incorporation of the variable time step method into GUARD should have no effect on the restart feature of the code; however, it has a direct impact on any post-processing program such as GRAFIX or PREP since the post-processing program will have to be aware that the output data is being recorded at a variable time step instead of a constant time step. Any post-processing program associated with GUARD will, in general, have to be modified to properly read and interpret the output files.

4.6 Eligible Interaction Elements and Panels

Previously, in the GUARD program, interaction force calculations were performed for every vehicle impact panel (including the front bumper)-barrier element combination. If there are L vehicle panels, M barrier impact rail elements and N contact lines per element, then a total of L x M x N sets of interaction calculations were performed every time step. In general, a large fraction of this total is unnecessary because a given vehicle panel may be nowhere near a specific barrier element and therefore could not possibly impact that part of the barrier. It is inefficient to allow interaction calculations to be performed for such "ineligible" panel/element combinations.

In an effort to increase the efficiency of interaction force calculations, tests were introduced into GUARD to determine the "impact eligibility" of vehicle panels and barrier elements. There are five tests in all, including an Initial Contact Test.

To verify the accuracy and effectiveness of these tests, a sample simulation problem was solved with and without the presence of the tests. The numerical results were identical for the two runs and it was determined that the presence of the tests saved 30 to 40 percent of interaction calculation time. However, because the interaction calculations constitute approximately 10 percent of the total running time in GUARD, only 3 to 4 percent savings in <u>total</u> run time were realized.

The logic of the eligibility tests is based on assumed panel and element numbering procedures. This introduces certain restrictions on the numbering and arrangement of vehicle impact panels. These restrictions should be clearly stated in the GUARD Users Manual.

The GUARD subroutines affected by insertion of the initial-contact and eligibility tests are indicated in the following table.

Table 1. Eligibility test subroutines.

TEST	SUBROUTINES AFFECTED			
	READIN	SOLVE	FREEFD	TSTCAR
Initial Contact	X	X	X	X
Eligibility				X

To implement the Initial-Contact Test, an additional parameter, ZR, has to be read in. It is to be read in on a separate "card," immediately after the current Card No. 4 (see GUARD Users Manual). The input format for ZR is E10.0 and it is to be included in Columns 1-10. This parameter locates the vertical, impact reference plane for the barriers.

4.7 Time Step Selection

The selection of a time step magnitude is generally dictated by consideration of both accuracy and numerical stability. To assist users of the GUARD program, guidance in the selection of time step was formulated and documented.

GUARD can treat either rigid or flexible barriers. For rigid barriers, the user is restricted to the use of a constant time step. For flexible barriers, a constant time step or a variable time step option can be chosen.

To assist in developing some guidelines for the selection of an appropriate constant time step magnitude for rigid barrier impact, a typical impact problem was solved in closed form using a simple vehicle model. This analysis indicated that a time step of 1 msec or less is required to obtain reasonable accuracy in the solution. This result is also borne out by past experience in solving numerous crash simulations with GUARD for rigid barriers. If practical, one can always try more than one time step for a given simulation to determine the sensitivity of the results to time step magnitude.

For flexible barriers, if the user opts to use a constant time step, it is recommended that the time step magnitude be no greater than the minimum natural period of the barrier model. This minimum period can be approximated using the newly incorporated feature described in section 4.4.

If the variable time step option is selected for a flexible barrier, the user must choose a value of tolerance parameter, ETSS, which governs the numerical accuracy of a solution time step. A range of values for this parameter was suggested based on limited experience gained with the use of the VTS feature in GUARD so far.

4.8 Errors Detected/Corrected

In performing work on the GUARD program, two groups of errors were detected. They had been introduced in the program on a previous contract.(20)

The first group of errors occurred when an attempt was made in reference 20 to correct the conversion of the representation of barrier nodal angular velocity and acceleration vectors from the "old" nodal coordinate system to the "new" nodal coordinate system every time step. The errors existed in subroutine FRCIN. They were corrected on this contract.

The second group of errors were produced in the program when the PREP and GRAFIX post-processing packages were added in reference 20. These packages will not work properly for any simulation where one or more restarts have been performed. This error was previously discussed in section 4.3 of this report. The error still exists in the program.

4.9 New Required Input Data

As a result of changes made to the GUARD program on this contract, some new additional information is required as input data to the program. These new requirements were discussed in the preceding sections and are summarized here. They were incorporated in a comprehensive Revised Input Guide for GUARD which appears in appendix 13 of this report. The new input data is:

- ITMSTP - Variable time step flag.
- ETSS - Tolerance parameter for variable time step calculation.
- TIMEMX - Maximum allowable solution time.
- ZR - Global z-coordinate of impact reference plane.
- IGRAPH- Flag for graphics output files for the GRAF and PREP programs.

Also, because of the incorporation of Panel and Element Impact Eligibility Tests in GUARD, there are now new limitations placed on the vehicle crush panel numbering scheme. These restrictions have been included in detail in a note in the Revised Input Guide for GUARD.

4.10 Application of Rigid Barriers

Work on this contract on the GUARD Computer Program was focused on simulation of vehicle impact with flexible barriers. A review was made of GUARD to determine what, if any, changes were required to ensure proper operation of the program relative to the utilization of the new enhancements and features when simulating rigid barrier impact. It was found that only minor changes are required to the code and to the Input Manual. These changes are described in appendix 15.

5. NARD PROGRAM CHANGES

5.1 Introduction

This contract specified that the CRUNCH program be modified to improve its accuracy and efficiency.(1) However, the FHWA subsequently chose its successor, NARD 1.0, as the version to be enhanced on this contract.(17) After examining the list of recommendations made with respect to the CRUNCH program, in section 2.3.3, FHWA selected the following areas to be addressed for NARD:

- Double precisioning.
- Barrier frequency calculation.
- Restart capability.
- Variable time step.
- Panel and element eligibility.
- Guidance on integration time step.
- Document explicit vs. implicit techniques.

The following sections summarize the effort performed in each of these areas. Detailed results are given in appendixes 14 through 18. These sections are followed by a section describing the incorporation of some auxiliary output files, a section discussing program errors discovered during the investigation and finally, ones summarizing new input data requirements and recommended notes for a revised users manual.

It should be noted that the changes described here to the NARD program were made only to the Implicit version of the code. The reason for this is that when the CRUNCH program was modified to develop the NARD 1.0 program, modifications comparable to those made to the Implicit version were not fully and properly made to the Explicit version.(1,17) Consequently, the Explicit version of NARD remains inoperable and changes made to NARD on this contract could only be performed on the Implicit version.

5.2 Double Precisioning

A test problem was run with NARD in both single precision and double precision modes. Double precisioning was achieved with the use of an IBM JCL parameter. The results revealed the following:

(1) There was approximately a 20 percent error in lateral position of the vehicle at the end of the run for the single precision mode and insignificant error in the longitudinal and vertical positions.

(2) Double precision required approximately 50 percent more user memory than single precision.

(3) Computer costs were approximately the same for the single and double precision runs at APL.

It was therefore recommended that double precisioning always be employed when operating on the IBM mainframe.

5.3 Barrier Natural Frequencies

It was desired to provide NARD with the ability to calculate an approximation to the highest natural frequency of a barrier model. This frequency governs determination of the largest time step possible to avoid numerical instability difficulties.

The method used for the Implicit version of NARD was the same as used for the GUARD program as described in appendix 8. It consists of calculating the "nodal frequencies" for each degree-of-freedom for each node in the model. The period associated with the highest frequency is determined and printed out. The calculation is performed during the pre-integration phase of the program. In order to make use of this calculation, the user should make a "pre-integration" run with the program, i.e., set MXSTEP=0 (no integration steps). It is suggested that the time step be set to a large value (e.g., 1000 sec). The output of this run (smallest nodal period) can be used as a guide in selection of a time step for the subsequent full integration run. It is recommended that a time step no larger than the smallest nodal period, or less, be used in the integration run.

To add the feature to the Implicit version it was necessary to:

(a) Include a new subroutine,called FREQ, in BARRIM. This routine is the same as the one in GUARD. A listing of it is given in appendix 8.

(b) CALL FREQ from subroutine SOLVE in BARRIM, before the integration phase coding but <u>after</u> the assembled stiffness matrix is adjusted for boundary conditions, i.e., after the CALL BOUND statement.

5.4 Restart Capability

A Checkpoint/Restart capability was given to the Implicit version of NARD. This feature allows a user to restart a previously run problem at any one of a number of prescribed points (times). An initial checkpoint run is made during which one or more "checkpoints" are recorded on a restart file. A checkpoint is a group of records written on the file at a specific step (time) in the run. The checkpoint contains information on the status of the model sufficient to allow one to use the information as the initial conditions for a subsequent restart run. The user may restart at any one of the checkpoints on the restart file and, in fact, may continue to record checkpoints on the restart file if desired. A second restart run may be made, and so on.

The checkpoint/restart capability was basically achieved by checkpointing all quantities in common blocks to the restart file. A new subroutine called RSTRT was constructed to both write and read the checkpoint information to the restart file. Logical Unit 20 is designated internally in RSTRT as the restart file. A listing of RSTRT is given in appendix 15. All variables in the common blocks in RSTRT are checkpointed to the restart file except those in common block RSTV. The three variables in that block (ICKPT, NINC and NSTRT) are specified by the user as newly required input relative to the checkpoint/restart feature.

When a checkpoint is written to the restart file, a statement to that effect, including the checkpoint number, is output to the print file so that the user knows how many checkpoints exist on the restart file for the problem at any time.

In addition to placing the new subroutine RSTRT in the VEHINTER module, the other changes to NARD for the checkpoint/restart capability were confined to the MAIN subroutine. The changes made to MAIN are detailed in appendixes 14 and 15.

It should be noted that, as in the case with GUARD, the GRAFIX and PREP post-processing programs should not be used for a restarted simulation because the NARD output files for those programs are not properly repositioned when restarting a simulation. This problem was not corrected on this contract because it was outside the scope of the contract to ensure that program changes were compatible with requirements of the GRAFIX and PREP programs.

Theoretically, one can save up to nearly 100 percent of run costs with the use of a restart feature. However, one should be selective as to the number of checkpoints specified during a simulation because a considerable amount of storage is required per checkpoint.

The user may elect any combination of checkpointing and/or restarting in a given computer run, i.e:

- No checkpointing or restarting.
- Checkpointing only.
- Restarting and checkpointing.
- Restarting only.

The user also specifies the vehicle time step frequency for which checkpoints are made. Different frequencies may be chosen for each of the three simulation phases (pre-contact, contact and post-contact). Restart/checkpoint information is still written to only one file (logical unit 20).

The new input quantities to the program required by the restart feature are as follows. They are to be read in as the first line of the input file. The format is 5I5.

ICKPT = checkpoint/restart flag

0 - no checkpointing or restart for the current run
1 - checkpointing only during run
2 - restarting and checkpointing during run
3 - restarting only during run

NINCPR = checkpoint increment during pre-contact phase (no. of vehicle time steps between checkpoints)

NINCCO = checkpoint increment during contact phase (no. of vehicle time steps between checkpoints)

NINCPO = checkpoint increment during post-contact phase (no. of vehicle time steps between checkpoints)

NSTRT = checkpoint number on restart file at which restart is to be made (if at all)

5.5 Variable Time Step

The Variable Time Step (VTS) algorithm incorporated into NARD was the same one as in GUARD and is presented in detail in appendix 10.

The subroutines which required changes were SOLVE, MAIN and ITERAT. Three new input parameters are required. They are "read" in subroutine MAIN as the third input "card" of the file. These parameters are:

- **ITMSTP** -- This is the variable time step flag. If it=0, then the time step is uniform throughout the contact phase of the simulation and is equal to the value of the initial time step, DT2, for the contact phase which is read in on the fourth card of the file. If ITMSTP = 1, the VTS feature is activated.

- **ETSS** -- This is the value of the desired uniform time step truncation error associated with the barrier solution. A detailed description of its theoretical definition is given in appendix 10. Some guidance of the selection of values for this parameter is provided in section 5.7.
- **TIMEMX** -- This is the maximum allowable solution time (sec), <u>after impact</u>. This was included to provide another control parameter to terminate a simulation. These are all discussed in section 5.10 of this report.

When the flag ITMSTP=1, the VTS feature is active, but only for the contact phase. In the pre- and post-contact phases the time steps are determined as before, i.e.,

<u>Pre-Contact Phase</u>

$$\Delta t \text{ for vehicle} = DT1$$

<u>Post-Contact Phase</u>

$$\Delta t \text{ for vehicle} = DT1$$
$$\Delta t \text{ for barrier} = IB*DT2$$

During the contact phase, if the VTS feature has been activated, the new time step, for both barrier and vehicle, is calculated according to the VTS algorithm described in appendix 10, but it is constrained to lie within the following bounds:

$$DELTL \leq \Delta t \leq DELTU$$

where

$$DELTL = 0.1 * TAUMIN$$

$$DELTU = 10.0 * TAUMIN$$

and TAUMIN is the minimum barrier nodal period computed by subroutine FREQ in the pre-integration phase of computations (see section 5.3).

These limits are not likely to be approached unless an unwise choice of ETSS is made by the user or unless extremely low or high rates of acceleration are experienced by the barrier.

If ITMSTP = 1, then during both the contact <u>and</u> post-contact phases, the barrier stiffness matrix will be updated <u>every</u> time step, irrespective of what the user specifies. This was forced, because otherwise the calculations during the contact phase would be incorrect.

To check out the new VTS feature, three VTS simulations and one constant time simulation were run for the same test problem. This provided some useful information on a suggested range of specified values of the tolerance parameter, ETSS. This is discussed further in section 5.7. However, this limited experience with the VTS feature did not provide clear cut evidence of the benefit, if any, obtained with the use of this feature. A more comprehensive study of constant time step versus VTS simulations must be performed before any generalizations can be made. One outcome of the work of this feature was that perhaps a smaller upper bound, DELTU, to the time step be used in the code, e.g., the use of DELTU = 2.0 * TAUMIN rather than DELTU = 10.0 * TAUMIN might very well result in better performance of the VTS feature.

As was the case with the VTS feature in GUARD, this feature, in NARD, is only applicable for flexible barrier impact.

No attempt was made to check on whether or not the GRAFIX and PREP programs can be used successfully to post-process a VTS simulation. However, it should be noted that the generic time history output files incorporated into the program on this contract (see section 5.9) <u>are</u> compatible with a VTS simulation.

5.6 Element and Panel Impact Eligibility

As was discussed in section 4.6 for the GUARD program, at a given instant in time in a simulation, the relative position of the vehicle and barrier may be such that it may not be possible for some vehicle panels to interact with some barrier elements. It would be inefficient to allow interaction calculations to be performed between all vehicle panels and all barrier elements. Therefore, quick "eligibility" tests are desired to determine those combinations of vehicle panels and barrier elements for which it is possible, geometrically, to be in contact.

A review of the NARD coding revealed that it already had extensive Eligibility Tests present. An attempt was made to improve upon some of the tests, with little success. The tests were ultimately left in their original form.

5.7 Guidance on Time Step Selection

The time step selection for the numerical solution of differential equations of motion is generally dictated by consideration of accuracy and numerical stability.

Accuracy of solution is affected by the introduction of truncation and round-off errors in the solution process. The truncation error corresponds to the error due to the approximation inherent in the integration method being employed. It is strongly dependent on the value of time step, being normally proportional to an integral power of time step size. Round-off error occurs simply because of the necessity of using finite arithmetic. Its magnitude is primarily a function of the computer hardware used. The per- step round-off error is fairly independent of time step, but the total solution round off error is dependent on the number of time steps, so that smaller time steps result in a greater number of steps in the solution and, therefore, greater round-off error. A compromise time step is generally required to balance truncation and round-off error effects.

Numerical instability of a solution procedure is basically caused by magnification of per-step truncation or round-off errors in computing subsequent steps. The stability characteristics of a procedure is dependent on several factors including the following:

- Formulation of procedure (e.g., implicit or explicit).
- Linearity of system.
- Integration method (recurrence formulas).
- Frequency of system.
- Time step.

The stability characteristics of the integration methods in NARD are not formally known; therefore, the selection of time step from a stability viewpoint must basically rely on experience.

The following parameters controlling time step selection have been recommended for NARD.

Table 2. Recommended time step sizes.

Type of Barrier	Time Step for	Time Step		
		Pre-Contact Phase	Contact Phase	Post-Contact Phase
Rigid	Vehicle	DT1	DT	DT1
Rigid	Barrier	---	--	---
Flexible	Vehicle	DT1	$\leq$TAUMIN (CTS) * (VTS)	DT1
Flexible	Barrier	---		DT2

DT1 $\leq$ 10 msec

DT = 1 to 5 msec

DT2 $\leq$ 1 msec and DT2 = DT1/n (n an integer)

---

* For VTS Option,
Initial Time Step $\leq$ TAUMIN
Tolerance Parameter: $0.006 \leq$ ETSS ≤ 0.012 inch

5.8 Implicit Versus Explicit Procedures

The terms Implicit and Explicit procedures refer to two different strategies commonly used to combine the differential equations of motion and the difference equations, associated with a given numerical integration method, to obtain a solution.

Advantages and disadvantages of the two procedures may be summarized as follows:

(1) Implicit

- Unconditionally stable (for linear problems), may use large time steps. However, for non-linear, this may lead to instabilities.
- Requires matrix inversion at every time step.
- Most suitable for problems
  - with low loading rates and/or
  - only low frequency response is of interest

(2) Explicit

- Does not require matrix inversion - no global stiffness matrix required.

- Only unconditionally stable, must use small time steps.

- Most suitable for problems
  - with high loading rates and/or
  - high frequency response is important

The CRUNCH program offers both solution procedures, the NARD program does not.(1,17) A comprehensive study of the use of explicit vs. implicit procedures should be made for typical crash scenarios to provide more definitive guidelines to users as to which procedure is most appropriate or efficient in this application.

5.9 Time History Files

Post-processing for NARD via PREP and GRAFIX in an interactive mode is very expensive on a mainframe computer. In addition, batch operation of these programs does not work well. Therefore, a supplemental output file capability was introduced in NARD, for vehicle and barrier response quantities, to allow time history plotting of these so-called "time-history" files using any plotting program that can read the unformatted structure of the time history files.

The vehicle and barrier time history files can both be independently turned on or off with an input on/off flag for them. Note that these output files can be constructed concurrently with those for the GRAFIX and PREP programs, or in lieu of them.

On this contract, all post-processing was accomplished using only the time-history files. These files were converted to a formatted mode, then downloaded to workstations for post-processing operations using an in-house plotting package.

The coding changes required to implement the vehicle time history file features in the program were confined to subroutine VEHICL. Details of changes made are described in appendix 14. For the barrier, all time history file coding was done in a new subroutine called BARHST, which is called from the first executable statement in subroutine OUTPUT. A listing of BARHST is provided in appendix 14.

No attempt was made to have the time history files support restart operations. The reason is two-fold. Firstly, the time history files were incorporated into NARD to provide an alternate means (to GRAFIX and PREP) of graphically verifying the changes made to the code on this contract. Graphics were not needed to validate the restart feature. Secondly, the addition of alternate time history files to the code was not a contractual item.

5.10 Notes for Users Manual

During the course of work on the NARD program on this contract, it became apparent that a few additional notes to the users manual are in order. They are as follows:

(a) Multi-Step Integration Factors

Multi-step integration factors (IB, IV, II) are input on the fourth line of the input file. The values of IB and IV must equal unity. It is recommended that II=1 also. The reason for this is that there are logical errors in the NARD code associated with the interaction force iteration procedure. This is discussed further in sections 5.11 and 6.2.

(b) Master-Slave Barrier Node Relationships

The master-slave barrier node relationships must be specified properly as indicated in the users manual. An error in these relationships for just one node can lead to a complete break-down of the solution as was demonstrated in appendix 16 of this report. It is recommended that some internal checks on data consistency be built into the code.

(c) Output for Barrier Slave Nodes

In the code, the translational displacements are updated every time step, but the corresponding velocities and accelerations are never updated; they remain zero, even though spaces are reserved for them in an array. Therefore, there is no point in requesting translational velocities or accelerations of slave nodes as barrier output quantities. This information is not currently contained in the users manual, but should be.

(d) Limits on Simulation Time and Steps

There are several input parameters which establish a limit on the simulation time or steps.

Parameters Read-In the Executive Module

NI = maximum allowable number of time steps during pre-contact phase

NC = during contact phase, the maximum number of consecutive time steps for which no contact occurs, before control is switched to the post-contact phase

TIMEMX = maximum allowable <u>barrier</u> solution time* after impact (sec)

<u>Parameters Read-In Vehicle Module</u>

CONTRL(1,1)= maximum <u>total vehicle</u> simulation time* (pre-contact, contact and post-contact)

CONTRL(1,5)= number of time steps between <u>vehicle</u> print steps <u>before</u> impact

CONTRL(2,3)= number of time steps between <u>vehicle</u> print steps <u>after</u> impact

<u>Parameters Read-In Barrier Model</u>

MXSTEP = maximum number of time steps allowed during <u>contact</u> phase

NPFREQ = <u>barrier</u> output <u>print</u> interval (steps)

---

* In the code there are two "clocks" recording simulation time, the vehicle clock and the barrier clock. The vehicle clock starts at zero from the very initial vehicle position. The barrier clock starts at zero at the time of initial impact. Therefore, during the contact phase, the vehicle and barrier "times" differ by the time it took the vehicle to initially impact the barrier starting from its initial position.

5.11 Errors Detected/Corrected

Two classes of errors were detected in the NARD program, those associated with (1) the soil/post interaction model and (2) the vehicle barrier interaction force iteration procedure.

(1) <u>Soil/Post Interaction Model Errors</u>

As described in section 3 of this report, there are a multitude of errors in the NARD 1.0 and 2.0 programs with respect to both the interpretation and implementation of the soil/post interaction model. These were not corrected on this contract. However, as reported in section 8, they were corrected in PCNARD which is the PC version of NARD 2.0.(21)

(2) <u>Vehicle/Barrier Interaction Force Iteration Procedure Errors</u>

These errors are discussed in section 6.2 and a description is given of how some were corrected and how the others can be "bypassed."

5.12 New Input Data Requirements

Some of the features added to the NARD program require new input data. These were discussed in detail in the preceding sections and appropriate appendixes, and are summarized here.

The new input quantities to the program required by the restart feature are as follows. They are to be read in as the first line of the input file from MAIN:

ICKPT = checkpoint/restart flag.
NINCPR = checkpoint increment during pre-contact phase.
NINCCO = checkpoint increment during contact phase.
NINCPO = checkpoint increment during post-contact phase.
NSTRT = checkpoint number on restart file at which restart is to be made.

Three new input parameters are required for the VTS feature. They are "read" in subroutine MAIN as the third input "card" of the input file. These parameters are:

ITMSTP = variable time step flag.
ETSS = accuracy tolerance parameter.
TIMEMX = maximum allowable solution time (sec), after impact.

Incorporation of output time history files for the vehicle and barrier response quantities require two input parameters:

NFPV -- vehicle time history flag
= 0, no vehicle time history file generated
= n, vehicle time history file generated on logical unit n, every time step.

NFPB -- barrier time history flag
= 0, no barrier time history file generated
= n, barrier time history file generated on logical unit n, every time step.

These parameters are read in the MAIN routine. Formerly the users manual specified that they be input as zero, because a time history output file feature had been planned but not fully implemented.

6. REVIEW OF NARD INTERACTION FORCE ITERATION PROCEDURE

6.1 Introduction

In order to plan the incorporation of the various new features into the code, it was necessary to review the interaction force iteration procedure in NARD. In doing so, two errors related to the procedure were discovered and inconsistency in the implementation of the procedure was noted. These are discussed in the following. More details of this review can be found in appendix 16.

6.2 Existing Errors

The first error has to do with the so-called multi-step integration, or subcycling, feature of the code. Since its inception the CRUNCH and its successor CRUNCHII (CWA/CRUNCH) have had the capability to perform temporal integration with, in general, different time steps for the barrier, vehicle and interaction modules during the contact phase of the simulation. This is specified by reading in the parameter DT2 as the base time step for the contact phase and the factors IV, IB and II which are to be applied to DT2 to determine the time integration increments for the vehicle, barrier and interaction modules respectively. When the interaction force iteration procedure was introduced into CRUNCH III (NARD), it was done in such a manner as to produce inconsistency in the times at which the interaction force between the vehicle and barrier is computed. For example, if DT2 is 1 msec and IV=5 and IB=2, (II is ignored in NARD even though the manual states that it is required) the interaction forces are computed every time step, the barrier is integrated every time step with a Δt of 2 msec and the vehicle is integrated every time step with a Δt of 5 msec. Therefore, the mutual interaction force is being applied to the barrier at 2, 4, 6,... msec after impact whereas it is being applied to the vehicle at 5, 10, 15,... msec after impact.

The multi-step error can be avoided by specifying IV=IB=1.

The second error found in NARD, and a consequence of the interaction force iteration procedure, is related to the vehicle "clock" and print-out frequency counter. With the current interaction procedure, the Vehicle Module is entered twice during each time step. Every time the module is entered, both the vehicle TIME and print-out counter are incremented. However, they should be incremented only once per time step, not twice. Consequently, the <u>vehicle</u> time which is printed out is incorrect and vehicle print-out frequency is twice what the user specifies it to be. This error was corrected.

6.3 Commentary and Recommendations

There is no formal documentation available on the interaction force iteration procedure in NARD. However, an informal explanation of the procedure was obtained.(22) According to reference 22, the intent of the procedure was to take "the average of the interaction forces... from the beginning and at the end of the time step." (In contrast to this, previous versions of CRUNCH used the interaction forces at the beginning of the time step to integrate both the vehicle and barrier equations.) However, NARD is <u>not</u> actually coded to use the average of the beginning and end-of-time-step interaction forces, but rather an average of "hybrid" interaction forces. A more consistent and straightforward implementation of the intended procedure was outlined and is presented in appendix 16. This procedure is conceptually much simpler than the current procedure in NARD.

Regardless of the specific form of the implementation of an interaction procedure, there is the greater question of whether such a procedure produces a great enough increase in computational accuracy to offset the great sacrifice made in efficiency. Since the three major computational modules (vehicle, interaction and barrier) are exercised twice during each time step, using the iteration procedure, the running time for a simulation is essentially doubled! There is no documentation available recording results of a study showing the increase in accuracy gained at the expense of efficiency lost.

7. BANDWIDTH MINIMIZATION

The GUARD and NARD programs employ a bandwidth solution method to solve the barrier model equations. The form of the equations are:

$$(F_{eff}) = K_{eff}(\Delta x)$$

where F_{eff} is the effective force vector, K_{eff} is the effective stiffness matrix and Δx is the unknown vector of incremental displacements for a given time step. The node numbers of the finite element mesh of the barrier are usually numbered such that the nonzero entries in the stiffness matrix are relatively clustered about the main diagonal with a "band" centered on the diagonal. The number of calculations required to solve the equations per time step, and hence solution time, is strongly dependent on the bandwidth of the stiffness matrix, the larger the bandwidth the longer the solution time. It is therefore desirable, for greater efficiency, to renumber (resequence) the nodes of the barrier model to achieve the minimum possible bandwidth.

A stand-alone computer program, BANDBAR, was developed to minimize the bandwidth of the stiffness matrix for the barrier model in GUARD and NARD. This was accomplished by adapting an available stand-alone computer program called BANDIT which is a matrix bandwidth reduction preprocessor written for u2e with the NASA structural analysis computer program, NASTRAN .(23,24) BANDIT is written in FORTRAN and uses the Cuthill-McKee strategy for resequencing node points.(25) The program was originally written for operation on CDC computers, but has been adapted by various users for operation on other computers.

A UNIVAC version of the BANDIT program was obtained and adapted to process GUARD and NARD input files instead of NASTRAN input. The resultant code, named BANDBAR reads an original full GUARD or NARD input file and creates a new GUARD (NARD) input file with resequenced barrier node numbers (corresponding to minimum bandwidth). The original or new file may be used as input to GUARD (NARD). The BANDBAR program was left as a stand-alone program; it was not incorporated into GUARD (NARD) because of its large size. One disadvantage in using BANDBAR as a stand-alone preprocessor is that the user must make note of the correspondence between the original and the new node numbering sequence of the barrier model. It should be noted that, because BANDBAR reads the entire input file for GUARD (NARD), any future input requirement changes made for them will necessitate corresponding changes to BANDBAR.

In order to verify the operational accuracy of the newly created BANDBAR program, it was used to process an "original" GUARD input file and create a corresponding "optimized" input file. Then both the original and optimized files were used as input to GUARD. The GUARD outputs associated with the original and optimized input files were compared to verify that the results were the same. The run times were also compared to determine how much solution time was reduced with the use of a reduced bandwidth barrier model. BANDBAR reduced the original bandwidth from 13 to 2 which resulted in a savings of approximately 50 percent in running time.

A sample test model for NARD was also processed with BANDBAR. The bandwidth was reduced from 5 to 4 with an associated savings in run time for the model of approximately 9 percent. This savings was obtained for a very short run (only 20 contact steps). Much greater savings would accrue for a longer run.

Details of the development of BANDBAR are described in appendixes 11, 12, 13 and 17. A full description of the output files from BANDBAR is given in appendix 17. Although that description is provided for a sample GUARD model, it is also applicable to a NARD model.

8. PCNARD PROGRAM CHANGES

As described in section 3 of this report, a state-of-the-art review was made of soil/post interaction. An interim report (appendix 6) documented the results of the review. One of the findings was that the implementation of the soil model in NARD 1.0 and NARD 2.0 possessed several errors both in interpretation and logic.(17,18)

A recommendation was made (appendix 7) to remove the soil model in NARD and replace it with the model in CWA/CRUNCH, the predecessor of NARD, so as to give FHWA a working soil model, albeit a relatively simple one, in the short-term while pursuing the development of more sophisticated models in the long-term.(1)

FHWA accepted the recommendation, but chose to have the soil model "transplant" performed on PCNARD rather than NARD.(21) PCNARD is essentially the PC version of NARD 2.0. This section summarizes the work performed in carrying out this recommendation. A detailed description of the work performed is given in appendix 19.

To replace the NARD soil model it was necessary to replace the following subroutines in PCNARD:

- SNSTIF.
- SSDATA.
- SFRCIN.
- SSNFRA.
- SSNFRD.

In addition, two other routines were modified to accommodate the new soil model; they are FRCIN and ITERAT. The changes to these routines were identified by liberal use of comment lines in the coding.

The new soil model requires a substantially different input format from the old NARD model. Appropriate changes to the users Mmnual to reflect this difference are documented in appendix 19 as well as necessary changes to the programmers' and engineering manuals of NARD.

Two problems were used to demonstrate the operability of the new soil model in PCNARD. The first problem consisted of the simulation of a 4500-lb pick-up truck impacting a Thrie Beam/Box Beam barrier transition section. This problem was supplied by FHWA, complete with an input file. Three runs were performed for this simulation. The first run, named "BOX," corresponded to the input file, as provided by the FHWA. It employed the NARD soil model, soil data as derived by others, and had the soil nodes located on the posts at grade level. In addition, five of the soil nodes were specified as fully fixed.

The second run for the BOX beam problem was named "BOXF" and the input was the same as the first run except that the "fixed" soil nodes were released to act as true soil nodes. In the third run, CWABOXF, the soil model was changed to the CRUNCH model with the depth of the soil nodes placed halfway between grade and the bottom of the posts beneath the soil. The soil data used was a transliteration of the original data so as to make it consistent with the CRUNCH soil model.

The results of the three BOX-BEAM runs are tabulated in table 3 which indicates a maximum vehicle acceleration of 12 g, a maximum post deflection of 15 inches and an abrupt vehicle redirection by the barrier for the "BOX" run. The other two runs produced considerably smaller vehicle accelerations and about twice as much post deflection. The vehicle redirection was much more moderate than in the "BOX" run. Note that the last two runs, which differed only in the soil model, produced similar results.

The second problem used for demonstration purposes was that of a standard W-Beam configuration. The original input data file was taken from reference 10. It corresponds to a 4700-lb vehicle impacting a standard, G4(1S) steel post, W-beam barrier at a 25 degree angle and 60 mph. Four simulations were conducted for this problem. The first run, "AGIBEAM," was made for the original input data and using the NARD soil model. In the other three runs, the CRUNCH soil model was used, with different combinations of soil data, depth of soil nodes and fixity of the lateral translational degrees-of-freedom of the soil nodes. The results are summarized in table 4. They indicate the following:

- The CRUNCH soil model produced somewhat smaller vehicle g-levels and a less abrupt vehicle redirection than did the NARD soil model.
- Different soil data produced small differences in vehicle acceleration, a 50 percent difference in post deflection and significantly different vehicle redirection severity.
- The change of assumption of the soil node lateral translational degree-of-freedom fixity, produced a very minor change in results.

A recommended procedure for deriving data necessary for input to the CRUNCH soil model in PCNARD was presented and a numerical example of the use of this procedure was given.

Table 3. Results of box-beam simulations.

Run Name	Soil Model Used	Soil Data Source	Depth of Soil Nodes (inches)	Nodes 23,28,33,38,43	Maximum Lateral Vehicle Acceleration (g)	Maximum Post Deflection (inches)	Vehicle Redirection
BOX	NARD	AGI	0.	Fixed	12.	15	Abrupt
BOXF	NARD	AGI	0.	Soil Nodes	5.	30	Moderate
CWABOXF	CRUNCH	AGI	20.4	Soil Nodes	3.5	33	Moderate

Table 4. Results of W-beam simulations.

Run Name	Soil Model Used	Soil Data Source	Depth of Soil Nodes (inches)	Soil Node Lateral Trans. DOF	Maximum Lateral Vehicle Acceleration (g)	Maximum Post Deflection (inches)	Vehicle Redirection
AGIWBEAM	NARD	AGI	0.	Active	13.	15.	Abrupt
AGIWBM2	CRUNCH	AGI	22.	Fixed	9.5	16.	Less Abrupt
CWAWBMA	CRUNCH	TTI	33.	Fixed	9.	24.	Moderate
CWAWBMB	CRUNCH	TTI	33.	Active	~9.	~24.	Moderate

REFERENCES

1. Chiapetta, R. L.
"CWA/CRUNCH Input Data Manual" (Revised)
Final Report, Contract DTFH61-83-P-10132, June 1985, unpublished (Contractor Report 4016)

2. Welch, R. E. and Pang, E.
"Validation and Application for the GUARD Computer Program"
Volume I: Program Validation; Volume II: Program Users' Manual
Final Report, Contract DOT-FH-11-9460, June 1984, unpublished (Contractor Report 4005)

3. Powell, G. H.
"BARRIER VII: A Computer program for the Evaluation of Automobile Barrier Systems"
Report FHWA-RD-73-51, Federal Highway Administration, Washington, DC

4. Swanson, J. A.
"ANSYS - Engineering Analysis System Users' Manual"
Swanson Analysis Systems, Inc., Elizabeth, PA

5. Shaeffer, H. G.
MSC/NASTRAN Primer Static and Normal Modes Analysis
Wallace Press, Inc.: Milford, New Hampshire, 1979

6. Hibbitt, H. H., et al
"ABAQUS-EPGEN Version 4-4, Volume I: Users' Manual and Volume II: Theoretical Manual"
EPRI Report NT-2709-CCM, October 1982

7. "MARC General Purpose Finite Element Program, Revision K.1"
MARC Analysis Research Corp., Palo Alto, CA, 1983

8. Bathe, K. J.
"ADINA, A Finite Element Program for Automatic Dynamic Incremental Nonlinear Analysis"
Massachusetts Institute of Technology Report 82448-1, December 1978

9. Belytschko, T.
"Transient Analysis" Structural Mechanics Computer Programs - Surveys, Assessments and Availability
The University Press of Virginia, 1974

10. Welch, R. E.
"Finite Element Analysis of Automotive Structures Under Crash Loadings"
IIT Research Institute Report J6321, May 1975

REFERENCES (Continued)

11. Winter, R., Pifko, A. and Armen,Jr., H.
"Crash Simulation of Skin-Frame Structures Using a Finite Element Code"
Paper 770484 presented at Soc. of Automotive Engineers, Business Aircraft Mtg., Wichita, Kansas, March 29-April 1, 1977

12. Bathe, K. J., Wilson, E. L. and Iding, R.
"NONSAP: A Structural Analysis Program for Static and Dynamic Response of Nonlinear Systems"
University of California, Structural Engineering Laboratory Report UC SESM 74-3, February 1974

13. MacNeal, R. H.
"The NASTRAN Theoretical Manual" (Level 12.1)
NASA-SP-221, September 1970

14. Welch, R. E.
DATAPROC Data Processor
Chiapetta, Welch & Associates, Ltd., Palos Hills, IL, November 1980

15. Shams, T., Nguyen, T. and Chi, M.
"GRAFIX - A Graphics Post Processor: Users' Manual"
Report FHWA/RD-83/037.1, Federal Highway Administration, Washington, DC, April 1983

16. Shams, T., Nguyen, T. and Chi, M.
"PREP - The Plotting and Reporting Program: Users' Manual"
Report FHWA/RD-83/037.7, Federal Highway Administration, Washington, DC, May 1983

17. "Numerical Analysis of Roadside Design (NARD)" Volume I: Users Manual,Report FHWA-RD-88-210; Volume II: Programmers Manual, Report FHWA-RD-88-212, Federal Highwway Administration, Washington, DC, Final Report, June 1988

18. Chou, C. C., Hancock, K. and Basu, S.
"NARD Version 2.0 - Volume I: Users Manual"
Report FHWA-RD-89-179, Federal Highway Administration, Washington, DC, July 1989

19. Martinez, J. E.
"Assessment of 2Vehicle Bumper-Guardrail Interaction"
Texas Transportation Institute Report RF3480, 1984

REFERENCES (Continued)

20. McHale, G. and Basu, S.
"GUARD Version 3.0: Users and Programmers Manual"
Report FHWA-89-RD-067, Federal Highway Administration,
Washington, DC, March 1989

21. McHale, G., Erinie, O. R. and Legere, J. F.
"Conversion of the NARD Computer Program to the PC Environment"
Report FHWA-RD-90-031, Federal Highway Administration,
Washington, DC, September 1989

22. Correspondence from S. Basu of Scientex Corporation to R. L. Chiapetta of Chiapetta, Welch & Associates, Ltd.; April 24, 1989

23. Everstine, Gordon C.
"The BANDIT Computer Program for the Reduction of Matrix Bandwidth for NASTRAN"
Department of the Navy, NSRDC, Report 3827, March 1972

24. McCormick, C. W., ed
<u>The NASTRAN Users' Manual</u>
NASA SP-222, September 1970

25. Cuthill, E. H. and McKee, J. M.
"Reducing the Bandwidth of Sparse Symmetric Matrices"
Proc. of the 24th National Conference ACM 1969, pp 157-172