September 27 - October 01, 2004, Bad Reichenhall, Germany

FINITE ELEMENT SIMULATION OF A WEAPON DETONATION IN A REINFORCED CONCRETE BRIDGE PIER

Dr. Kent T. Danielson, Dr. James L. O'Daniel, and Dr. Mark D. Adley

Network Computing Services/Army HPC Research Center, ERDC
US Army Engineer Research and Development Center
3909 Halls Ferry Road, CEERD-GM-I, Vicksburg, MS 39183 USA

This paper presents the modeling effort and associated predictions for a numerical simulation of a penetrating weapon detonating in a reinforced concrete bridge pier. The basic objective of the simulation is to assess the level of damage in the pier as a result of the payload detonation. This analysis was performed with the finite element code, *ParaAble*, developed jointly by the Army High Performance Computing Research Center (AHPCRC) and the U.S. Army Engineer Research and Development Center (ERDC). The weapon was modeled as an uncased cylinder of tritonal with assumed quarter symmetry and the entire model consisting of approximately 3.5 million elements and nodes. All materials are explicitly modeled with eight-noded hexahedral elements. The reinforcing steel was modeled using an Elasto-Plastic material model with increased strengthening for strain rate effects. The tritonal was modeled with a JWL equation of state and, due to the high level of confinement, the properties were obtained using an assumption of an ideal explosive with 100% reaction of all materials. A programmed burn model ignited the top center of the cylinder. The concrete was modeled with a microplane constitutive theory developed jointly by ERDC and Professor Zdeněk Bažant, Northwestern University. Included with the microplane concrete model in *ParaAble* is a damage output variable that permits the inspection of relative damage and the prediction of failed regions. The damage variable is determined by a modified Holmquist-Johnson-Cook approach that relates damage to levels of inelastic strain increment and pressure. *ParaAble* was designed to efficiently perform large-scale analyses by exploiting parallel computing via Message Passing Interface (MPI) calls. For a 10 millisecond simulation, this analysis used about 30 cpu hours on 512 processors (~15000 hours total) of a Cray T3E-1200 system or 10 cpu hours on 256 processors of a Compaq AlphaServer SC45 system.