Progress in Nuclear Energy 66 (2013) 1–12

Contents lists available at SciVerse ScienceDirect

Progress in Nuclear Energy

journal homepage: www.elsevier.com/locate/pnucene

Progress in the development of an implicit steady state solution in the coupled code TRACE/PARCS

M.S. Ellis [a,b,*], J. Watson [a,b,*], K. Ivanov [b]

[a] Applied Research Laboratory, The Pennsylvania State University, University Park, PA 16802, USA
[b] Department of Mechanical and Nuclear Engineering, Nuclear Engineering Program, The Pennsylvania State University, 230 Reber Building, University Park, PA 16802, USA

ARTICLE INFO

Article history:
Received 6 November 2012
Received in revised form
8 February 2013
Accepted 12 February 2013

Keywords:
Implicit coupling
TRACE
PARCS
Multiphysics

ABSTRACT

This paper describes the implementation of an implicit steady state solution method in the TRAC/RELAP Advanced Computational Engine (TRACE) thermal-hydraulics system code and Purdue Advanced Reactor Core Simulator (PARCS) code with the goal of improving solution stability and efficiency. The implicit steady state solution method has been implemented within the framework of the existing psuedo-transient solution method in TRACE and includes time-dependent thermal-hydraulic and heat transfer equations and time-independent neutron diffusion equations. The implicit solution method uses Newton's method to solve the thermal-hydraulic, heat transfer, and neutron diffusion equations during each psuedo-time step with an analytic construction of the Jacobian matrix. The linear system associated with each iteration of Newton's method is solved using a preexisting LU decomposition algorithm in TRACE. The implicit steady state solution was evaluated using two different meshes overlaid on a two-phase pipe model closely matching a boiling water reactor hydraulic channel. The implicit solution method reproduces the correct steady state solution for varying time step sizes for each mesh. An evaluation of the CPU runtime required to complete a steady state calculation using the implicit method shows that the well-developed and optimized explicit solution method currently requires less CPU runtime than the implicit solution method which has yet to be optimized. These results direct future development of the implicit solution method towards optimization strategies to reduce CPU runtime.

1. Introduction

The simulation of nuclear reactor systems under steady state and transient conditions is a particularly challenging task because of the multiple physical processes occurring simultaneously within a reactor. Earlier attempts to model the multiphysic characteristics of nuclear reactors relied on splitting the different physical processes into separate computational units, a technique referred to as Operator Splitting (OS) (Ragusa and Mahadevan, 2009). In reactor analysis the OS technique typically consists of a hydraulic component used for calculating core flow, heat transfer component used for determining structure temperatures, and a neutronics component used for calculating the neutron population in the core. Unfortunately, the OS technique is inherently nonlinearly inconsistent and is inferior to numerical methods that resolve the nonlinearities of the physical system through more rigorous methods (Ragusa and Mahadevan, 2009). The limitations of OS techniques were investigated with an analytical approach for several nonlinear equation sets by Knoll et al. (2003), and it was shown that OS methods can lead to a loss of solution accuracy when large time steps are utilized. The loss of solution accuracy and resulting need for smaller time steps is the main catalyst for investigating improved OS methods and nonlinearly consistent methods.

As it applies to time-dependent problems, OS splitting is typically synonymous with explicit time-integration techniques where the field of unknowns is reduced and linearized through the use of previous time step solutions for nonlinear coefficients and principle unknowns. Unfortunately, explicit time-integration techniques are not unconditionally stable and are limited to smaller time steps because of stability constraints and first-order temporal accuracy (Watson, 2010). However, even with the aforementioned time step size restrictions, explicit time integration techniques are commonly used to solve reactor analysis problems (Ragusa and Mahadevan, 2009).

In its simplest form, the explicit time integration method does not include an exchange of solution values within a time step but

[*] Corresponding authors. Applied Research Laboratory, The Pennsylvania State University, University Park, PA 16802, USA. Tel.: +1 814 8650040; fax: +1 814 865 3287.
E-mail addresses: mse5011@arl.psu.edu (M.S. Ellis), jkw104@arl.psu.edu (J. Watson).

0149-1970/$ – see front matter © 2013 Elsevier Ltd. All rights reserved.
http://dx.doi.org/10.1016/j.pnucene.2013.02.009

Nomenclature

Independent variables

t	Time
x	Cartesian coordinate direction
y	Cartesian coordinate direction
z	Cartesian coordinate direction

Other variables

C_p	Specific heat at a constant pressure
k	Thermal conductivity
P	Pressure
q'''	Volumetric heat-generation rate
T	Temperature
α	Vapor volume fraction
ρ	Microscopic density
Σ	Macroscopic cross section
ϕ	Neutron flux
a	Neutron leakage coefficient
k_eff	Effective multiplication factor
k_s	Shifted effective multiplication factor
v	Neutron speed
h	Neutronics node length
χ	Fraction of delayed or prompt neutrons
β	Delayed-neutron fraction
Q	Total model (core) rated thermal power
M	Neutronics migration matrix
F	Neutronics fission matrix
Ψ	Fission Source
nr	Number of heat structure nodes in given axial plane
κ	Average energy release per fission
ν	Average number of neutrons produced per fission
C	Delayed neutron precursor concentration
J	Neutron Current
λ	Delayed neutron precursor decay constant or eigenvalue
D	Neutron Diffusion Coefficient
$\widehat{D}$	Nodal Coupling Correction Coefficient
$\tilde{D}$	Nodal Coupling Coefficient
Δu	Distance between neutronics nodes
R	Neutron flux scaling factor
W_i	Power deposition weighting factor

Superscripts

m	Neutronics node number
i	Heat Structure node number
j	Hydraulic node number
$m+$	Neighboring neutronics node number in the positive direction
$m-$	Neighboring neutronics node number in the negative direction
$j+$	Neighboring hydraulic node number in the positive direction
$j-$	Neighboring hydraulic node number in the negative direction
$i+$	Neighboring heat structure node number in the positive direction
$i-$	Neighboring heat structure node number in the negative direction
$m \pm l_u$	Neighboring neutronics node number either the positive or negative direction u
M	All other neutronics nodes other than cell m
J	All other hydraulic nodes other than cell j
I	All other heat structure nodes other than cell i
i'	All other heat structure nodes coinciding with cell m other than i
n	Current time level
$n+1$	New time level
k	Current iterate
$k+1$	Next iterate

Subscripts

a	Noncondensable-gas component
d	Delayed
f	Fuel
g	Gas field
g	Neutron energy group
g'	Neutron energy group other than neutron energy group g
$g'g$	From energy group g' to energy group g
l	Liquid field
k	Delayed neutron precursor group
p	Prompt
u	Any coordinate direction
m	Mixture

passes information at the end of the time step (Ragusa and Mahadevan, 2009). Another common implementation of the explicit time integration method uses a staggered update scheme where the split computational units are executed sequentially and converged solution information is passed only in the forward direction (Ragusa and Mahadevan, 2009). The latter implementation of the explicit time integration technique is the method utilized in TRACE and PARCS, and a schematic of the flow of data is shown Fig. 1.

The inherent time step size limitation and potential error of the explicit time integration technique is the primary impetus for including fully implicit time integration methods in reactor analysis codes. In implicit time integration techniques all of the independent parameters coupling the multiple physical processes are solved for simultaneously at the time step of interest instead of the separated convergence indicative OS methods. Early investigations that pertained to the development of fully implicit two-phase flow equations showed improved time step size capability and increased accuracy compared to existing operator-splitting and explicit time-integration techniques (Mousseau, 2004; Frepoli et al., 2003; Mousseau, 2005). Other investigations that sought to develop implicit solution techniques for entire sets of hydraulic, heat transfer, and neutronics equations reported varied, but promising, improvements in solution accuracy and maximum time step sizes or an increase in solution efficiency for a steady state solution (Kastanya and Turinsky, 2005; Mousseau, 2006; Mousseau and Pope, 2007; Ragusa and Mahadevan, 2009; Watson, 2010).

A major obstacle impeding the widespread implementation of fully implicit methods is the inefficiencies associated with solving all of the nonlinear equation sets simultaneously. To solve the nonlinear system of equations, some variant of Newton's method is typically used because of its second-order convergence rate when a good initial solution estimate is available (Keyes et al., 2006). The efficacy of Newton's method is largely dominated by the efficiency of constructing and inverting the Jacobian matrix. The most direct method is to construct the Jacobian matrix using analytical expressions for equation derivatives and solving the resulting system of linear equations using a direct or iterative solver. Because