Copyright © 2011 Tech Science Press CMC, vol.21, no.3, pp.217-235, 2011

Parallel Finite Element Method and Time Stepping Control for Non-Isothermal Poro-Elastic Problems

Wenqing Wang[1], **Thomas Schnicke**[2] **and Olaf Kolditz**[3]

Abstract: This work focuses on parallel finite element simulation of thermal hydraulic and mechanical (THM) coupled processes in porous media, which is a common phenomenon in geological applications such as nuclear waste repository and CO2 storage facilities. The Galerkin finite element method is applied to solve the derived partial differential equations. To deal with the coupling terms among the equations, the momentum equation is solved individually in a monolithic manner, and moreover their solving processes are incorporated into the solving processes of nonisothermal hydraulic equation and heat transport equation in a staggered manner. The computation task arising from the present method is intensive if the method is applied to model a real geological application. Therefore, we present a parallel finite element method and a time stepping method with PI (proportional and integral feedback) automatic control to improve the computation efficiency. For parallel computing, the domain decomposition method is unitized to partition both computation tasks of the equation assembly and the linear solve, and the establishment of a global system of equations is thoroughly avoided. Moreover, an object-oriented concept of sparse matrix and iterative linear solver for large scale parallel and sequential simulation is developed. By simulating a real application with THM coupled processes, we show that the present parallel finite element method works fine for both monolithic and staggered scheme within coupling iterations, and furthermore we show the efficiency of the present method by the speedup we have achieved in the simulation.

Keywords: Parallel finite element method, adaptive time stepping with automatic control, non-isothermal multi-phase flow, poro-elasticity

[1] Helmholtz Centre for Environmental Research - UFZ, Leipzig, Germany. Corresponding author. Email: wenqing.wang@ufz.de

[2] Helmholtz Centre for Environmental Research - UFZ, Leipzig, Germany

[3] Helmholtz Centre for Environmental Research - UFZ, Leipzig, Germany, Technical University of Dresden, Dresden, Germany

1 Introduction

For many geo-engineering problems such as the safety assessments of geothermal reservoirs, CO2 storage and nuclear waster disposal (Doughty and Pruess, 2004; Stephansson, Hudson, and Jing, 2004; Alonso, Alcoverro), the time dependent partial differential equations (PDEs) arising from them are highly nonlinear coupled and have to be solved numerically (Lewis and Schrefler, 1998; Rutqvist, Börgesson, Chijimatsu, Kobayashi, Nguyen, Jing, Noorishad, and Tsang, 2001; Rutqvist, Barr, Datta, Gens, Millard, Olivella, Tsang, and Tsang, 2005; Sanavia, Pesavento, and Schrefler, 2006; Wang and Kolditz, 2007). One of the most frequently used and the most robust numerical methods for solving such highly nonlinear coupled PDEs is the finite element method (Lewis and Schrefler, 1998). Usually, the computation task of the finite element analysis of real geo-engineering problems is expensive. There are several ways to improve the computational efficiency, e.g. more efficient numerical algorithms, optimization of memory management in the code, and parallelization techniques. In the numerical analysis of time dependent thermo-hydraulic processes in porous media, the time stepping is a crucial issue for numerical stability and computational efficiency. Practically, the fixed time step size does not often satisfy the stability and efficiency requirements in solving problems that exhibit complexity in geometry and nonlinearity in material properties. Therefore, adaptive time stepping methods taking high-order integration into account have been developed and are widely applied (Hairer and Wanner, 1996). Among the available adaptive time stepping methods, the well-known techniques for prediction of the time step size h are e.g. Courant number approach based on Courant-Friedrichs-Lewy condition (Courant, Friedrichs, and Lewy, 1967) for the finite difference method, primary variable based prediction (e.g. (Minkoff and Kridler, 2006; Ouyang and Tamma, 1996)) and local error control methods (e.g. (Gustafsson, 1991, 1994; Hairer and Wanner, 1996)). The local error control methods especially those based on theoretical control ideas are problem independent for any numerical methods for ODEs (Gustafsson, 1991, 1992; Söderlind and Gustafsson, 1997; Hairer and Wanner, 1996). For nonlinear equations, the theory based automatic controls such as P (proportional feedback) or PI (proportional and integral feedback) permit stable and efficient time stepping (Söderlind and Gustafsson, 1997; Hairer and Wanner, 1996) for numerical solver. In addition to the efficient numerical algorithms, parallel finite element computing provides the most powerful speed-up (Salinger, Xiao, Zhou, and Derby, 1994; Topping and Khan, 1996; Fujisawa, Inaba, and Yagawa, 2003; Tezduyar and Sameh, 2006; Wang, Kosakowski, and Kolditz, 2009) if the required hardware is available. The present work is subjected to incorporate the parallel finite element method and the adaptive time stepping with automatic control for the analysis of thermal hydraulic and mechanical coupled processes in porous media.

For the adaptive time stepping, we present an approach of PI (proportional and integral feedback) automatic time stepping for modeling the problems with different coupled physical processes. With the present time stepping approach, each process uses the time step size predicted by the PI control of itself to guarantee the stability of the simulation of each process under coupling. For the parallel finite element method computing, the local and global assembly, solving linear equation system are the most time consuming parts. The discretization of the weak form of an initial-boundary-value problem in the finite element space results in linear equation systems with sparse matrices. More large grid posed to the finite element analysis means a more large sparse stiffness matrix, and also definitely means more computational expense. There are many references about the algorithms to solve sparse linear equation systems (e.g. (Ortega, 1988; Saad, 2003)). Regarding to the parallel linear solver, there are popular standalone and portable packages available such as PETSc (Balay, Buschelman, Gropp, Kaushik, and McInnes, 2007), Aztec (Tuminaro, Heroux, Hutchinson, and Shadid, 1999), PSBLAS (Filippone and Colajanni, 2000), AMG (specifically for multigrid method) (Henson and Yang, 2002) to mention a few. Despite the availability of these packages, there are still good reasons to develop specific solvers for memory management, easy code maintaining and developing for new parallel algorithm. Based on the domain decomposition, the present work develops an parallel finite element method in the frame work of object oriented programming, with which the sparse matrices and the associated iterative linear solvers can be handled in a cheap way for parallel and sequential finite element simulation of mutli-field problems. The present methods are implemented in the framework of OpenGeoSys (http://www.opengeosys.net), a scientific open source object-oriented parallel FEM simulator (Wang and Kolditz, 2007; Wang, Kosakowski, and Kolditz, 2009), and they are verified with a real application in this work.

2 Theoretical background of multi-field problem

We treat the partially saturated porous media as multi-phase system composed of constitutes with the voids of the solid skeleton filled with water and gas. We assume that capillary pressure p^c, gas pressure p^g, absolutely temperature T, and displacement $\mathbf{u}$ are primary variables to describe the state of the porous media. The governing equations are given hereafter.

2.1 Mass balance equation

Consider two fluid flow in porous media, e.g liquid (denoted by l) and gas (denoted by g). For each phase in multi-phase fluid flow in deformable porous media, the governing equation of flow field can be derived (Kolditz and de Jonge, 2004; Wang, Rutqvist, Görke, Birkholzer, and Kolditz, 2011) by taking account of deformation in mass balance:

$$\frac{\partial}{\partial t}\left(nS^g\rho_k^g + nS^l\rho_k^l\right) + \nabla\cdot\left(\mathbf{J}_k^g + \mathbf{J}_k^l\right) + \rho_k\frac{\partial}{\partial t}(\nabla\cdot\mathbf{u}) = Q_k \tag{2.1}$$

where k denotes the components of the fluid, e.g air ($k = a$) and water ($k = w$), S is saturation, ρ stands for phase density, n is the porosity, $\mathbf{J}$ is total flux, $\mathbf{u}$ is the displacement vector, and ρ_k defined by

$$\rho_k = \begin{cases} \rho_w^l S^l + \rho_w^g(1-S^l), & k = w \\ \rho_a^g(1-S^l), & k = a \end{cases}$$

For any phase $\gamma = (g,l)$, an advection vector $\mathbf{J}_{A_k^\gamma}$ and a diffusion vector $\mathbf{J}_{D_k^\gamma}$ makes up a total flux, i.e

$$\mathbf{J}_k^\gamma = \mathbf{J}_{A_k^\gamma} + \mathbf{J}_{D_k^\gamma} \tag{2.2}$$

According to the Darcy's equation, the advective part of the total flux can be written as

$$\mathbf{J}_{A_k^\gamma} = -\rho_k^\gamma\frac{\mathbf{k}k_{rel}^\gamma}{\mu^\gamma}\left(\nabla p^\gamma - \rho^\gamma\mathbf{g}\right) \tag{2.3}$$

where $\mathbf{k}$ is the intrinsic permeability, k_{rel}^γ is the relative permeability of the phase, and μ^γ is the viscosity.

The diffusion part of the total flux is given by Fick's law as

$$\mathbf{J}_{D_k^\gamma} = -nS^\gamma\rho^\gamma\mathbb{D}_k^\gamma\nabla\left(\frac{\rho_k^\gamma}{\rho^\gamma}\right) \tag{2.4}$$

where $\mathbb{D}_k^\gamma$ is a diffusion parameters in the terms of tensor. Since $\rho^\gamma = \rho_a^\gamma + \rho_w^\gamma$, we have

$$\mathbf{J}_{D_w^\gamma} + \mathbf{J}_{D_a^\gamma} = \mathbf{0} \tag{2.5}$$

under the assumption of $\mathbb{D}_a^\gamma = \mathbb{D}_w^\gamma$

Consider water-air mixture. We expand the mass balance equation (2.1) with the flux defined in equations (2.2) based upon the above equations (2.2, 2.3, 2.4). For water component, the diffusion part of the total flux takes the form

$$\mathbf{J}_{D_w^l} = -nS^l\rho^l\mathbb{D}_w^l\nabla\left(\frac{\rho_w^l}{\rho^l}\right), \quad \mathbf{J}_{D_w^g} = -nS^g\rho^g\mathbb{D}_w^g\nabla\left(\frac{\rho_w^g}{\rho^g}\right) \tag{2.6}$$

Obviously, $\mathbb{D}_w^l = \mathbf{0}$. Therefore, the mass balance equation for water component can be written as follows

$$\frac{\partial}{\partial t}\left(nS^g\rho_w^g + nS^l\rho_w^l\right) - \nabla\cdot\left[\rho_w^l\frac{\mathbf{k}k_{rel}^l}{\mu^l}\left(\nabla p^l - \rho^l\mathbf{g}\right)\right]$$
$$-\nabla\cdot\left[\rho_w^g\frac{\mathbf{k}k_{rel}^g}{\mu^g}\left(\nabla p^g - \rho^g\mathbf{g}\right)\right] - \nabla\cdot\left[nS^g\rho^g\mathbb{D}_w^g\nabla\left(\frac{\rho_w^g}{\rho^g}\right)\right] + \rho_k\frac{\partial}{\partial t}\left(\nabla\cdot\mathbf{u}\right) = Q_w \tag{2.7}$$

Since the capillary pressure p^c is chosen as one of the two unknowns of equation (2.1) and $S^g = 1 - S^l$, equation (2.7) becomes

$$n(\rho_w^l - \rho_w^g)\left(\frac{\partial S^l}{\partial T}\frac{\partial T}{\partial t} + \frac{\partial S^l}{\partial p^c}\frac{\partial p^c}{\partial t}\right) + \left(\rho_w^l S^l + \rho_w^g(1-S^l)\right)\frac{\partial}{\partial t}\left(\nabla\cdot\mathbf{u}\right)$$
$$+(1-S^l)n\left(\frac{\partial\rho_w^g}{\partial T}\frac{\partial T}{\partial t} + \frac{\partial\rho_w^g}{\partial p^g}\frac{\partial p^g}{\partial t} + \frac{\partial\rho_w^g}{\partial p^c}\frac{\partial p^c}{\partial t}\right)$$
$$-\nabla\cdot\left[\rho_w^l\frac{\mathbf{k}k_{rel}^l}{\mu^l}\left(\nabla(p^g - p^c) - \rho^l\mathbf{g}\right)\right]$$
$$-\nabla\cdot\left[\rho_w^g\frac{\mathbf{k}k_{rel}^g}{\mu^g}\left(\nabla p^g - \rho^g\mathbf{g}\right)\right] - \nabla\cdot\left[n(1-S^l)\rho^g\mathbb{D}_w^g\nabla\left(\frac{\rho_w^g}{\rho^g}\right)\right] = Q_w \tag{2.8}$$

Similar to the previous procedure, the diffusion part of the total flux of air component can be written as

$$\mathbf{J}_{D_a}^{\ l} = -nS^l\rho^l\mathbb{D}_a^l\nabla\left(\frac{\rho_a^l}{\rho^l}\right), \quad \mathbf{J}_{D_a}^{\ a} = -nS^g\rho^g\mathbb{D}_a^g\nabla\left(\frac{\rho_a^g}{\rho^g}\right) \tag{2.9}$$

The density shift from air component to liquid ρ_a^l is very small and can be omitted anyway. Therefore, we can assume $\mathbf{J}_{D_a}^{\ l} \approx 0$. As a consequence, the mass balance equation for air component is derived as:

$$\frac{\partial}{\partial t}\left(nS^g\rho_a^g\right) - \nabla\cdot\left[\rho_a^g\frac{\mathbf{k}k_{rel}^g}{\mu^g}\left(\nabla p^g - \rho^g\mathbf{g}\right)\right]$$
$$-\nabla\cdot\left[nS^g\rho^g\mathbb{D}_a^g\nabla\left(\frac{\rho_a^g}{\rho^g}\right)\right] + \rho_a^g(1-S^l)\frac{\partial}{\partial t}\left(\nabla\cdot\mathbf{u}\right) = Q_a \tag{2.10}$$

Expending the temporary derivative term of equation (2.10) yields

$$-n\rho_a^g\left(\frac{\partial S^l}{\partial T}\frac{\partial T}{\partial t} + \frac{\partial S^l}{\partial p^c}\frac{\partial p^c}{\partial t}\right) + \rho_a^g(1-S^l)\frac{\partial}{\partial t}\left(\nabla\cdot\mathbf{u}\right)$$
$$+(1-S^l)n\left(\frac{\partial\rho_a^g}{\partial T}\frac{\partial T}{\partial t} + \frac{\partial\rho_a^g}{\partial p^g}\frac{\partial p^g}{\partial t} + \frac{\partial\rho_a^g}{\partial p^c}\frac{\partial p^c}{\partial t}\right)$$
$$-\nabla\cdot\left[\rho_a^g\frac{\mathbf{k}k_{rel}^g}{\mu^g}\left(\nabla p^g - \rho^g\mathbf{g}\right)\right] - \nabla\cdot\left[n(1-S^l)\rho^g\mathbb{D}_a^g\nabla\left(\frac{\rho_a^g}{\rho^g}\right)\right] = Q_a \tag{2.11}$$

Mass balance equations (2.8) and (2.11) are exactly the same as that described in the paper by Sanavia, Pesavento, and Schrefler (2006).

ρ_w^g is the so called water vapor density. As an example, we give its expression based on the Claperon equation of perfect gas and Dalton's law as follows

$$p_a^g = \rho_a^g RT/M_a,\, p_w^g = \rho_w^g RT/M_w$$
$$p_g = p_a^g + p_a^w,\, \rho^g = \rho_a^g + \rho_w^g \tag{2.12}$$

In the partially saturated zone, the equilibrium water vapor pressure p_{gw} can be derived from the Kelvin-Laplace equation

$$p_w^g = p_{gws} \exp\left(-\frac{p^c M_w}{\rho_w^l RT}\right) \tag{2.13}$$

Hereby, we use empiric water vapor saturation function, p_{gws}, as

$$p_{gws} = \frac{10^{-3} p_c M_w}{RT} \exp(19.84 - 4975.9/T) \tag{2.14}$$

2.2 Energy balance equation

We consider the convective transport, i.e. the transport of heat by flow. The energy balance equation of the porous media is

$$\begin{aligned}\rho C_p \frac{\partial T}{\partial t} &+ \rho^w C_p^w \frac{\mathbf{k} k_{rel}^w}{\mu^w}\left(-\nabla p^g + \nabla p^c + \rho^w \mathbf{g}\right)\nabla T \\ &+ \rho^g C_p^g \rho^g \frac{\mathbf{k} k_{rel}^g}{\mu^g}\left(-\nabla p^g + \rho^g \mathbf{g}\right)\nabla T + \nabla(K_e \nabla T) + Q_{\mathrm{T}} = 0\end{aligned} \tag{2.15}$$

where ρ is the density of porous media, C_p is the effective specific heat capacity, K_e is the heat conductivity of the porous media.

2.3 Deformation process

Deformations in porous media can be described by the momentum balance equation in the terms of stress under the assumption of solid grains being incompressible (Lewis and Schrefler, 1998),

$$\nabla \cdot \left[\boldsymbol{\sigma} - \sum_{\gamma}^{\gamma=(g,l)} S^\gamma p^\gamma \mathbf{I}\right] + \rho \mathbf{g} = 0, \tag{2.16}$$

where $\boldsymbol{\sigma}$ denotes the effective stress of porous medium, $\mathbf{I}$ stands for identity tensor. In the present study, the traditional sign convention for stress and fluid pressure is used. Density of porous media consists of the portion contributed by liquid l and by the portion contributed of solid as $\rho = n(\rho_w + \rho_a) + (1-n)\rho^s,$. Displacement $\mathbf{u}$ is the primary variable to be solved by substituting the constitutive law for stress-strain behavior

$$\begin{aligned} \boldsymbol{\sigma} &= \mathbb{C}\left(\boldsymbol{\varepsilon} - \alpha_T \Delta T \mathbf{I}\right) \\ \boldsymbol{\varepsilon} &= \frac{1}{2}(\nabla\mathbf{u} + (\nabla\mathbf{u})^{\mathrm{T}}) \end{aligned} \tag{2.17}$$

with $\mathbb{C}$, a forth order material tensor, α_T the thermal expansion coefficient, $\boldsymbol{\varepsilon}$ the strain, and ΔT the temperature increment. Superscript T means the transpose of matrix.

3 Finite element method

Although mass balance equation (2.8, 2.11), energy balance equation (2.15) and momentum equation (2.17) are different in equation types, we can represent them by a general time dependent partial differential equation as

$$m\dot{u} + \mathscr{L}u = f, \forall x \in \Omega \subset \mathbb{R}^n, n = 1,2,3 \tag{3.1}$$

where m is the mass parameter, $\mathscr{L}$ denotes a linear operator, f is the source/sink term, and u is the unknown field function to be solved. The concept of the finite element method is to solve a weak form of equation (3.1) numerically. Assuming $v \in V := C^1(\Omega) \cap C^0(\bar{\Omega})$ is a test function, the weak form of equation (3.1) can be given as follow:

$$(m\dot{u}, v) + (\mathscr{L}u, v) = (f, v) \tag{3.2}$$

where $(,)$ is a bilinear form. We solve equation (3.2) based on the Galerkin method. By choosing a finite element subspace $V_h \subset V$, which is typically spanned by a convenient set of shape functions $h_i (i = 1, 2, \cdots, n)$ with n the number of element nodes, we can solve the equation (3.2) by searching an approximate solution $u_h \in V_h$ such that

$$(m\dot{u}_h, v) + (\mathscr{L}u_h, v) = (f, v), \quad \forall v \in V_h \tag{3.3}$$

For equations of the present model problem, the derived element matrices and vectors can be classified into following types (Table 3.1)

Type	Name	Equations
$\int_\Omega \boldsymbol{N}_1^{\mathrm{T}} \mathscr{M} \boldsymbol{N}_1 \,\mathrm{d}\Omega$	Mass matrix M	(2.8), (2.11), (2.15)
$\int_\Omega (\boldsymbol{N}_1)^{\mathrm{T}} \mathscr{M} \nabla \boldsymbol{N}_1 \,\mathrm{d}\Omega$	Advection matrix	(2.8), (2.11), (2.15)
$\int_\Omega (\nabla \boldsymbol{N}_1)^{\mathrm{T}} \mathscr{M} \nabla \boldsymbol{N}_1 \,\mathrm{d}\Omega$	Laplace matrix K	(2.8), (2.11), (2.15)
$\int_\Omega \mathbf{B}^{\mathrm{T}} \mathscr{M} \mathbf{B} \,\mathrm{d}\Omega$	Tangential matrix	(2.17)
$\int_\Omega \mathscr{M} \mathbf{B}^{\mathrm{T}} \mathbf{m} \boldsymbol{N}_1 \,\mathrm{d}\Omega$	Displacement coupling matrix	(2.17)
$\int_\Omega \mathscr{M} \boldsymbol{N}_1^{\mathrm{T}} \mathbf{m}^{\mathrm{T}} \mathbf{B} \,\mathrm{d}\Omega$	Pressure coupling matrix	(2.17))
$\int_\Omega Q \boldsymbol{N}_1 \,\mathrm{d}\Omega \quad , \int_\Omega Q \boldsymbol{N}_2 \,\mathrm{d}\Omega$	Source term vector	(2.8), (2.11), (2.15),(2.17)
$\int_\Gamma q \boldsymbol{N}^\Gamma \,\mathrm{d}\Gamma$	Neumann vector	(2.8), (2.11), (2.15),(2.17)

Table 3.1: Matrix and vector types with material variable $\mathscr{M}$

where $\mathscr{M}$ are a process-specific material functions, $\mathbf{B} = \mathscr{L}\boldsymbol{N}_2$ is so called strain-displacement matrix, $\mathbf{m}$ is mapping vector defined as.

$$\mathbf{m} = \begin{cases} (1,1,1,0,0,0), & \text{3D problem} \\ (1,1,1,0), & \text{2D problem} \end{cases}$$

Such classification provides an easy way of implementation (Wang and Kolditz, 2007). After assembly of all element level matrices and vectors, we obtain a system of linear equations to be solved:

$$Au_h = b \tag{3.4}$$

The stiffness matrix A is s sparse type matrix , which is populated primarily with zeros.

4 Adaptive time stepping with automatic control

In this section, we describe the development and the implementation of the adaptive time stepping for thermo-hydraulic processes in porous media.

4.1 Temporal discretization

We use the backward Euler or implicit method to approximate the solutions of ordinary differential equations (3.3). Considering a relaxation for the Laplacian related term, we can discretize equations (3.3) as follows

$$M(u_h|_{n+1} - u_h|_n)/\Delta t + K\left[\alpha u_h|_{n+1} + (1-\alpha) h_u|_n\right] = f \tag{4.1}$$

where f is the right hand side including the contributions from the source/sink term, Neumann boundary conditions and the advection term, n and $n+1$ indicate the previous and current time steps,respectively, Δt denotes time step size $t_{n+1} - t_n$, $\alpha \in [0,1]$ is a relaxation parameter.

The fixed point or the Picard method is adopted to linearize equation (4.1). Therefore, for each Picard iteration $k+1$, the solutions of equations (4.1) are given by

$$u_h|_{n+1}^{k+1} = \left[M^k/\Delta t + \alpha K^k\right]^{-1} (f^k + M^k u_h|_n/\Delta t - (1-\alpha)K_p^k u_h|_n) \tag{4.2}$$

where k indicates the previous Picard iteration. Hereafter, for the sake of consistency of symbols with that used in the theory of automatic control, we denote time step size Δt as h.

4.2 Automatic control of time stepping

We employ the PI (proportional and integral feedback) (Gustafsson, Lundh, and Söderlind, 1988) classic time control method, which provides a stable and efficient time stepping for the numerical solution of PDEs. The idea behind PI control is the elementary local error control theory, i.e. the next time step size, h_{n+1}, can be predicted by the local error estimation e_n as (Gustafsson, Lundh, and Söderlind, 1988; Hairer and Wanner, 1996; Söderlind, 2002)

$$\frac{h_{n+1}}{h_n} = \eta \tag{4.3}$$

where η is the constant of time step size factor. In the present work, we apply the PI automatical control provided in (Hairer and Wanner, 1996) to calculate η. Denoting $\zeta = \varepsilon/e_n$ as the error excess, the time step size factor takes the form

$$\eta = \max\left(c_1, \frac{\min\left(c_2, \sqrt[4]{\zeta}\right)}{c_f}\right) \tag{4.4}$$

where c_1, c_2, c_3 are the constants, ε is the error of solution u_{n+1}^{k+1} in current nonlinear iteration, $e_n = e_A + e_r max(u_{n+1}^{k+1}, u_n)$ with an absolute tolerance e_A and a relative tolerance e_r.

Alternatively, if previous time step $n-1$ is accepted, the calculation of time step factor (4.4) can be combined with the predictive control presented in (Gustafsson, 1992) as

$$\begin{aligned} \eta_g &= \max\left(c_1, \min(c_2, \frac{h_{n-1}}{h_n}\sqrt[4]{\zeta^2/\max(\zeta_{n-1}, 10^{-2})}/c_f)\right) \\ \eta &= \max(\eta, \eta_g) \end{aligned} \tag{4.5}$$

4.3 Time stepping for coupled processes

Frequently staggered scheme are applied for the numerical simulations of the coupled problems. By this approach the dimension of the system of equations (degree of freedom) can be reduced. In regard to adaptive time stepping, the time step sizes of different processes predicted by PI control can be very different. To enable different process uses different step size and meanwhile keep time synchronization is therefore essential for numerical simulations of coupled process in a staggered manner. In the present study, we present a scheme that uses PI control to predict time step size for each coupled process and in addition synchronizes time step for data output at given times.

We consider a coupled problem has m processes, a set of specific times $t_l, l = 1, \cdots, l_{max}$ for output or some other purposes, and the predicted time step size of process i is h^i_{n+1}, $i = 1, \cdots, m$ after time t_n. In current step t_{n+1}, a time step size h is selected such that h is the smallest predicted time step size of all coupled processes as

$$h = \min(h^i_{n+1}, i = 1, \cdots, m)$$

Concerning to each process i, we set a variable h^i_c to accumulate the step size h such as

$$h^i_c = \begin{cases} h^i_c + h & \text{if } h_c^{<} h^i_{n+1}, \\ h^i_c & \text{else} \end{cases} \tag{4.6}$$

This means if the predicted step size of process h^i is larger than the selected time step size h, we skip the action on the process and increase the accumulation, h^i_c, with h, and we only perform the analysis of the process if the accumulation is larger than or equals to h^i_{n+1}.

5 Parallelization

The geometric parallelism is considered, with which all CPU nodes of a parallel machine run the same code. To partition the computational task of the finite element method, we use the domain decomposition approach. In the present parallel strategy, the parallel computing incorporated with PI automatic time step control takes the following steps:

- Before time stepping

  1. Partition the finite element method into several subdomains.
  2. Construct topological data of subdomain concurrently by each CPU.

- In each time step for individual physical equation

    1. Build local system of equations of subdomain concurrently by each CPU using local tropologic data of the mesh.
    2. Solve linear equation by iterative solver, where the product of the stiffness matrix and vectors is performed on subdomain by each CPU, and MPI communication is involved to collect norms of the result vectors before convergence and to collect the solution after convergence (Wang, Kosakowski, and Kolditz, 2009).
    3. Predict time step size by using PI control.

- After time step, synchronize times from each coupled equations by the approach described in section 4.3.

With the present parallel method, we successfully avoid building a global system of equations for each physical equations.

5.1 Sparsity and mesh topology

Sparsity of matrices results arises from the mesh topology of the finite element method. Making use of sparsity features is very important for both reduction of memory consumption for matrix storage and fast data access to matrix entries for computational efficiency. Shape functions of the finite element method are local and they have small support, i.e.

$$h_i(\xi_k) = \begin{cases} 1, & i = k \\ 0, & i \neq k \end{cases}, \; k = 1, 2, \cdots, ne \tag{5.1}$$

where i and k stand for node indices, ξ_k are local coordinates. This leads to a fact that most of the entries of the stiffness matrix A of the linear equation arising from equation (3.3) are zero ones. Moreover, the sparse pattern of a stiffness matrix from the finite element method is determined by the topology of the element mesh.

The local (element-by-element) assembly procedure results in following dependencies of the global stiffness matrix:

- *Each row of the global matrix is related to each node index.*
- *Non-zero entries of a row are determined by the nodes of all elements that are connected to a node, which is associated with this row.*

5.2 Storage scheme of sparse matrix and object oriented programming

The storage scheme of sparse matrices is aimed at minimization of memory consumption and maximization of computational efficiency. To this purpose, only the non-zero entries are stored and used for algebraic matrix operations. This can be easily done by using the topology of the finite element mesh (Section 5.1). Proven storage types are sparse vector, coordinate storage, compressed row storage, compressed column storage, compressed diagonal storage, jagged diagonal storage (JDS), block compressed row storage and skyline storage schemes (Saad, 2003). Since the JDS is very efficient at the cost of a gather/scatter operation, we utilize this type of matrix storage for the code implementation.

The sparse matrix pattern arising from the finite element method is resulting from mesh topology. For multi-field problems, however, the individual OPDEs may have different degree of freedom of the unknowns (i.e. scalar or vector quantities). Moreover, different interpolation schemes are necessary for the finite element method in order to guarantee numerical accuracy and stability. This type of multi-field problems leads to more complex sparsity pattern. The idea is that simply uses the number of element nodes as a shift to access sparse matrix entries. This prompts that the sparse matrix is better to be abstracted into two individual data objects: (1) for sparse matrix structure and (2) for matrix data entries as well as related algebraic operations. An important and novel aspect of the present work is the development of sparse matrix concept for multi-field problems. Moreover, a large flexibility for coupling schemes is introduced based on the object-orientated matrix concept.

- `SparseTable`: holds the sparse pattern of the stiffness matrix and two overloaded constructors, for sequential with the global mesh topology as an argument and parallel simulations with the subdomain mesh topology as an argument, respectively.

- `SparseMatrix`: contains values of matrix entries according to sparsity and degree of freedom per node, as well as provides methods algebraic matrix operations,

- `Solver`: Krylov-type subspace solver using the matrix operations provided by `SparseMatrix`.

6 Example

To investigate the computational efficiency of the present parallel scheme and time stepping approach for coupled PDEs, we conduct parallel simulations of the THM coupled processes in the FEBEX type repository defined in the DECOVALEX-THMC project (Barr, Birkholzer, Rutqvist, and Sonnenthal, 2004; Rutqvist, Barr, Birkholzer, Chijimatsu, Kolditz, Liu, Oda, Wang, and Zhang, 2008; Rutqvist, Barr, Birkholzer, Fujisaki, Kolditz, Liu, Fujita, Wang, and Zhang, 2009). The parallel computing involves a number of CPUs up to thirty.

Fig. 6.1 depicts the near field model of the FEBEX type repository, in which the specified observation points are given. 6.1. Fig. 6.1 depicts the definition of the benchmark, The linear element is adopted for hydraulic and heat transport equations, while the quadratic element is used for deformation equation. This makes the dimension of linear equation of deformation much larger than that of the other coupled equations. Detailed description of the material parameters and model set-up can be found in the paper by Wang and Kolditz (2007).

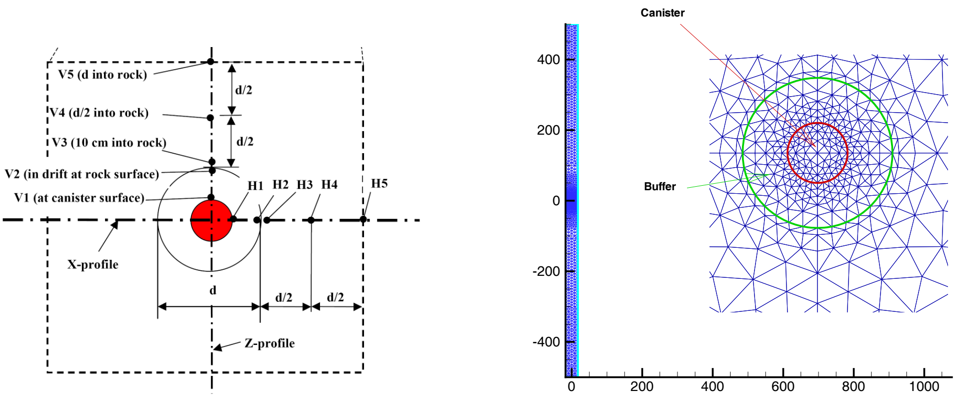

Figure 6.1: Left: Near field of the repository model(Barr, Birkholzer, Rutqvist, and Sonnenthal, 2004). Right: Finite element mesh

For PI time stepping, all parameters are assigned with their default values except $e_A = 10^{-10}$, $e_r = 10^{-4}$ for hydraulic and heat transport equation. Since deformation is elastic and time only exists in coupled terms, the time step size for the deformation equation is specified manually. The simulation duration is one million year, and the initial time step sizes are set as 0.001 year for hydraulic equation and 0.01 year for heat transport equation.

We measure the speed-up for this simulation based on wall clock time measurements on the SUN X4600 cluster. The achieved speedup of total time consuming by the linear solver and by the entire simulation is shown in Fig. 6.2. A nearly optimal speed-up can be achieved for up to 16 CPU nodes, then the parallel efficiency goes down. We can see that there is no significant speedup for linear solver of linear element approach owing to very few CPU time is consumed by this solver. Compared to the quadratic element approach, the linear element approach gives small dimension of linear equation, and its time consuming is dominated by inter-node communication. We also can find that the most important factor in speedup is the behavior of the linear solver for quadratic element. Since there is inter-computer-node communication in the parallel solver algorithm, an increasing number of iterations causes larger communication latency. Therefore, the overall speed-up is affected by the number of solver iterations. The inter-computer-node communication takes place on subdmain border nodes during each solver iteration and once on all mesh nodes after solver convergence is achieved. The number of subdmain border nodes is monotonically increasing with the increase of sub-domain numbers. If the number of subdmain border nodes becomes close to the number of the whole mesh nodes, the communication between border nodes will limit the speedup. In other words, if we decompose a mesh into too many sub-domains, we can not achieve a further speed-up for the parallel simulation.

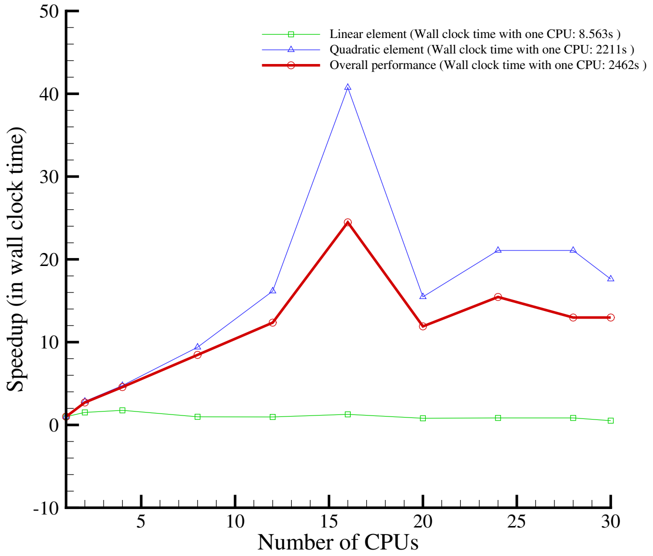

Figure 6.2: Speed-up of parallel finite simulation.

In the parallel computing, the PI control based algorithm for coupling equations performs a perfect time stepping for this thermal hydraulic coupled nonlinear problem. The simulation is finished in only 66 steps with only one rejected step, and its total non-linear iterations is 660. Fig. 6.3 shows variations of time step size of the two coupling equations in the synchronized time. It shows that time step sizes

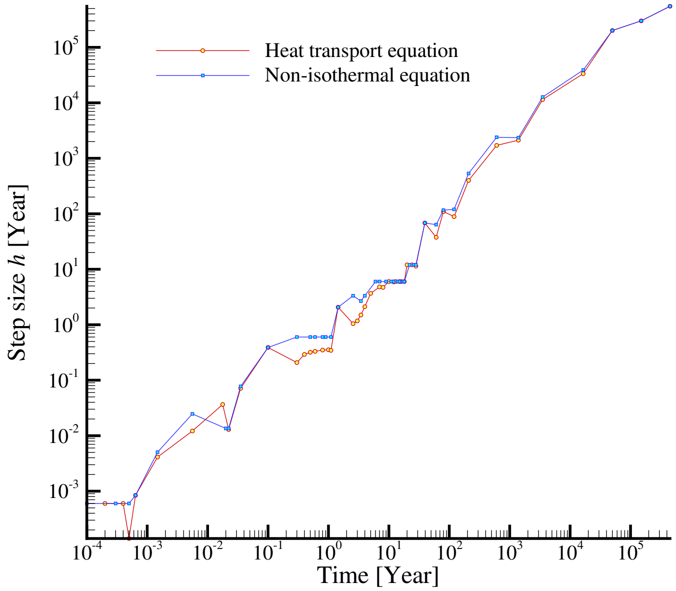

Figure 6.3: Time steps of non-isothermal Richard's problem

increase to a certain number until 1year and do not change much from 1 year to 100 year. This is because the temperature experiences its peak value, and consequently the water in the bentonite undergoes vaporizing. This phenomena is duplicated numerically by the present simulation, which is demonstrated by Fig. 6.4 for temperature and saturation variations at observation point V3 (see Fig. 6.1), which locates in the bentonite buffer just above the top of the canister.

The present temperature results are not only compared with that of our previous study but also with those by using TOUGH and ROCMAS by LBNL, and this is demonstrated in Fig. 6.4.

7 Conclusions

In the present study, we presented a high performance computing scheme for modeling the non-isothermal poro-elastic problems, which represent thermal hydraulic and mechanical coupled processes in porous media. The scheme combines an adaptive time stepping with automatic control and a parallel finite element method. The adaptive time stepping allows individual PDE of physical process use its own predicted time step size. While the parallel finite element scheme based on the domain

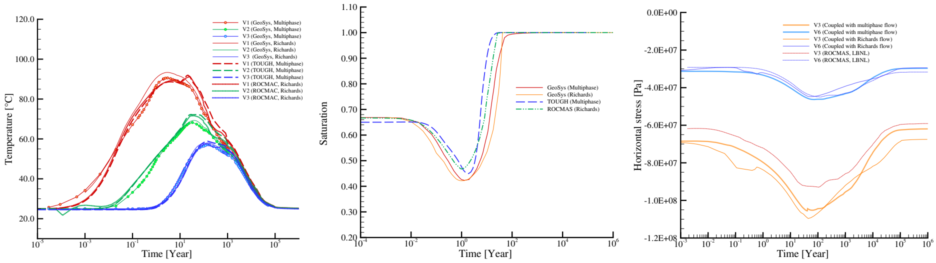

Figure 6.4: Left: Temperature variation at points V1, V2 and V5. Middle: Temperature variation at points V1. Right: Variation of horizontal stress

decomposition method and it is realized with MPI implementation. The sparsity of the derived stiffness matrix and its relates linear solver are handled in the objected oriented manner such that the memory consuming and inter-node communication are moderated. The present scheme was successfully verified by modeling a well known thermal hydraulic and mechanical coupled problem arising from nuclear waster disposal engineering.

Acknowledgement: The numerical model development was conducted in the framework of the international DECOVALEX project. The funding from the Federal Institute for Geosciences is highly acknowledged (Dr. Shao). This work is part of the PoF research initiative of the Helmholtz Association within the Environmental Engineering and Geothermal Technology programs.

References

Alonso, E. E.; Alcoverro, J.; Coste, F.; Malinsky, L.; Merrien-Soukatchoff, V.; Kadiri, I.; Nowak, T.; Shao, H.; Nguyen, T. S.; Selvadurai, A. P. S.; Armand, G.; Sobolik, S. R.; Itamura, M.; Stone, C. M.; Webb, S. W.; Rejeb, A.; Tijani, M.; Maouche, Z.; Kobayashi, A.; Kurikami, H.; Ito, A.; Sugita, Y.; Chijimatsu, M.; Borgesson, L.; Hernelind, J.; Rutqvist, J.; Tsang, C. F.; Jussila, P. (2005): The FEBEX benchmark test: case definition and comparison of modelling approaches. *International Journal of Rock Mechanics and Mining Sciences*, vol. 42, no. 5-6, pp. 611–638.

Balay, S.; Buschelman, K.; Gropp, W.; Kaushik, D.; McInnes, Lois Curfmanand Barry, F. (2007): *PETSc users manual.* PETSc home page, http://www-unix.mcs.anl.gov/petsc/petsc-as/, 2007.

Barr, D.; Birkholzer, J.; Rutqvist, J.; Sonnenthal, E. (2004): Draft description for DECOVALEX-THMC task D: Long-term permeability/porosity changes in edz and near field, due to THM and THC processes in volcanic and crystalline-bentonite systems. Technical report, Earth Sciences Division, Lawrence Berkeley National Laboratory, USA, 2004.

Courant, R.; Friedrichs, K.; Lewy, H. (1967): On the partial difference equations of mathematical physics (english translation of the 1928 german original). dowload availabe at: http://www.stanford.edu/class/cme324/classics/courant-friedrichs-lewy.pdf. *IBM Journal*, vol. March, pp. 215–234.

Doughty, C.; Pruess, K. (2004): Modeling supercritical carbon dioxide injection in heterogeneous porous media. *Vadose Zone Journal*, vol. 3, pp. 837–847.

Filippone, S.; Colajanni, M. (2000): PSBLAS: a library for parallel linear algebra computation on sparse matrices. *ACM Transactions on Mathematical Software*, vol. 26, no. 4, pp. 527–550.

Fujisawa, T.; Inaba, M.; Yagawa, G. (2003): Parallel computing of high-speed compressible flows using a node-based finite-element method. *International Journal for Numerical Methods in Engineering*, vol. 58, no. 3, pp. 481–511.

Gustafsson, K. (1991): Control theoretic techniques for stepsize selection in explicit runge-kutta methods. *ACM Transactions on Mathematical Software*, vol. 17, no. 4, pp. 533–554.

Gustafsson, K. (1992): *Control of Error and Convergence in ODE Solvers, Ph.D. thesis,Lund Institute of Technology*. PhD thesis, Lund Institute of Technology, Lund, Sweden, 1992.

Gustafsson, K. (1994): Control-theoretic techniques for stepsize selection in implicit runge-kutta methods. *ACM Transactions on Mathematical Software*, vol. 20, no. 4, pp. 496–517.

Gustafsson, K.; Lundh, M.; Söderlind, G. (1988): A PI stepsize control for the numerical solution of ordinary differential equations. *BIT*, vol. 28, no. 2, pp. 270–287.

Hairer, E.; Wanner, G. (1996): *Solving Ordinary Differential Equations II: Stiff and Differential-Algebraic problems, 2nd Revised Edition*, volume 14 of *Springer Series in Computational Mathematics*. 2nd rev. ed., Spinger-Verlag, Berlin.

Henson, V. E.; Yang, U. M. (2002): BoomerAMG: a parallel algebraic multigrid solver and preconditioner. *Applied Numerical Mathematics*, vol. 41, no. 1, pp. 155–177.

Kolditz, O.; de Jonge, J. (2004): Non-isothermal two-phase flow in low-permeable porous media. *Computational Mechanics*, vol. 33, no. 5, pp. 345–364.

Lewis, R. W.; Schrefler, B. A. (1998): *The Finite Element Method in the Static and Dynamic Deformation and Consolidation of Porous Media (Second Edition).* Wiley, New York, USA. 508 pp.

Minkoff, S. E.; Kridler, N. M. (2006): A comparison of adaptive time stepping methods for coupled flow and deformation modeling. *Applied Mathematical Modelling*, vol. 30, no. 9, pp. 993 – 1009.

Ortega, J. M. (1988): *Introduction to Parallel & Vector Solution of Linear Systems.* Plenum Press, New York, NY, USA.

Ouyang, T.; Tamma, K. K. (1996): On adaptive time stepping approaches for thermal solidification processes. *International Journal of Numerical Methods for Heat & Fluid Flow*, vol. 6, no. 2, pp. 35–50.

Rutqvist, J.; Barr, D.; Birkholzer, J. T.; Chijimatsu, M.; Kolditz, O.; Liu, Q.; Oda, Y.; Wang, W.; Zhang, C. (2008): Results from an international simulation study on coupled thermal, hydrological, and mechanical processes near geological nuclear waste repositories. *Nuclear Technology*, vol. 163, no. 1, pp. 101–109.

Rutqvist, J.; Barr, D.; Birkholzer, J. T.; Fujisaki, K.; Kolditz, O.; Liu, Q.; Fujita, T.; Wang, W.; Zhang, C. (2009): A comparative simulation study of coupled thm processes and their effect on fractured rock permeability around nuclear waste repositories. *Environmental Geology*, vol. 57, no. 6, pp. 1347–1360.

Rutqvist, J.; Barr, D.; Datta, R.; Gens, A.; Millard, A.; Olivella, S.; Tsang, C. F.; Tsang, Y. W. (2005): Coupled thermal-hydrological-mechanical analyses of the Yucca Mountain drift scale test - Comparison of field measurements to predictions of four different numerical models. *International Journal of Rock Mechanics and Mining Sciences*, vol. 42, no. 5-6, pp. 680–697.

Rutqvist, J.; Börgesson, L.; Chijimatsu, M.; Kobayashi, A.; Nguyen, T. S.; Jing, L.; Noorishad, J.; Tsang, C. F. (2001): Thermohydromechanics of partially saturated geological media - Governing equations and formulation of four finite element models. *International Journal of Rock Mechanics and Mining Sciences*, vol. 38, pp. 105–127.

Saad, Y. (2003): *Iterative Methods for Sparse Linear Systems.* SIAM, second edition.

Salinger, A. G.; Xiao, Q.; Zhou, Y.; Derby, J. J. (1994): Massively parallel finite element computations of three-dimensional, time-dependent, incompressible flows in materials processing systems. *Computer Methods in Applied Mechanics and Engineering*, vol. 119, no. 1–2, pp. 139–156.

Sanavia, L.; Pesavento, F.; Schrefler, B. A. (2006): Finite element analysis of non-isothermal multiphase geomaterials with application to strain localization simulation. *Computational Mechanics*, vol. 37, no. 4, pp. 331–348.

Söderlind, G. (2002): Automatic control and adaptive time-stepping. *Numerical Algorithms*, vol. 31, no. 1–4.

Söderlind, G.; Gustafsson, K. (1997): Control strategies for the iterative solution of nonlinear equations in ODE solvers. *SIAM Journal on Scientific Computing*, vol. 18, no. 1, pp. 23–40.

Stephansson, O.; Hudson, J.; Jing, L.(Eds): *Coupled Thermo-Hydro-Mechanical-Chemical Processes in Geo-Systems: Fundamentals, Modelling, Experiments, and Applications*, Geo-Engineering Book Series 2, London, UK, 2004. Elsevier. 852 pp.

Tezduyar, T. E.; Sameh, A. (2006): Parallel finite element computations in fluid mechanics. *Computer Methods in Applied Mechanics and Engineering*, vol. 195, no. 13-16, pp. 1872–1884.

Topping, B. H. V.; Khan, A. I. (1996): *Parallel finite element computations.* Saxe-Coburg Publications, Edinburgh.

Tuminaro, R.; Heroux, M.; Hutchinson, S.; Shadid, J. (1999): Official Aztec user's guide: Version 2.1, 1999.

Wang, W.; Kolditz, O. (2007): Object-oriented finite element analysis of thermo-hydro-mechanical (THM) problems in porous media. *International Journal for Numerical Methods in Engineering*, vol. 69, no. 1, pp. 162–201.

Wang, W.; Kosakowski, G.; Kolditz, O. (2009): A parallel finite element scheme for thermo-hydro-mechanical (THM) coupled problems in porous media. *Computer & Geosciences*, vol. 35, no. 8, pp. 1631–1641.

Wang, W.; Rutqvist, J.; Görke, U.; Birkholzer, J.; Kolditz, O. (2011): Non-isothermal flow in low permeable porous media: a comparison of richards' and two-phase flow approaches. *Environmental Earth Sciences*, vol. 62, no. 6, pp. 1197–1207.