# Parallel Stochastic Dynamic Programming Applications: Spatial and State Finite Elements *

Floyd B. Hanson† and John J. Westman

Laboratory for Advanced Computing
University of Illinois at Chicago
851 S. Morgan; M/C 249
Chicago, IL 60607-7045

E-Mail: hanson@math.uic.edu

**Abstract**

The optimal control of groundwater quality in an environment with stochastic input of contaminants is treated by numerical methods. The introduction of pollution contaminants to an aquifer is modeled as the random rare events of a Poisson process. The optimal policy is a schedule of pumping rates for an array of pumping wells that best reduces the contamination to levels which are in accordance with groundwater quality standards. Finite element representation is used independently for the aquifer geometric coordinates and the state variables for both flow and contaminant concentration. The finite element representation leads to a discretized PDE of stochastic dynamic programming in the finite element state space. The large dimensionality of this discretized PDE makes its computation a grand challenge, suitable for a massively parallel processor.

*Keywords: Stochastic Dynamic Programming, Parallel Computational Control,Groundwater Pollution Management*

## 1. Introduction

The large scale computational requirements of dynamical programming, deterministic or stochastic, are called the *Curse of Dimensionality*, since the requirements grow exponentially with the state space dimensions as powers of the state mesh size. This is characteristic of *Grand Challenge* problems [7]. The methods developed in these investigations would be useful in aerospace, biology, chemistry, computers, engineering, finance, physics and resource problems.

For stochastic dynamic programming (SDP) in continuous time, state and control, numerical PDE methods have been modified for vector multiprocessing and the nonstandard characteristics of the Bellman's equation [9]. Further high performance computing advances using vector supercomputing and massively parallel computing have been made by this group [16, 17]. Resource problems have been solved with up to 5 states and 32 finite difference nodes per state on both Cray and Connection Machines. Larger scale problems can be solved with recent hardware and software advances. Our general formulation treats many of the problems considered as important in *Future Directions in Control Theory* [6], such as computational control, feedback control, nonlinear control, and stochastic control.

For deterministic problems, ODE shooting methods can be used for the Hamiltonian equations derived from the minimum principle instead of dynamic programming. Also differential dynamic programming (DDP) for deterministic problems can reduce the computational demand by quadratic expansions in differentials about nearby state trajectories [10]. Dantas de Melo and co-workers [5] use Cray vectorization and multitasking techniques on discrete dynamic programming control problems.

Here the focus is on stochastic dynamic programming for a pollution remediation application in an uncertain environment. This application has both state and spatial dependence which greatly increase the computational demands. Replacing finite differences by finite elements can lead to less storage requirements, due to greater accuracy of finite elements. The storage saving property was demonstrated in [4] for a two-dimensional application on a Cray supercomputer. The construction of procedures for state and spatial finite elements makes the choice of mesh data structures very important. The vector data structure for state finite differences developed in [8] is extended to finite elements here. The use of finite element methods have a long history in groundwater research. Pinder and Frind [15] compared finite element and finite difference methods for stationary groundwater problems, finding little difference in accuracy between the methods.

Yakowitz [18] gives a thorough summary of the application of many variants of dynamic programming to a large number of groundwater problems. Here, the groundwater problem is the remediation of randomly discharged pollution into a confined underground aquifer. An optimal pumping and treatment policy is sought for pumping out concentrations of pollution until environmental standards are met, if feasible. This is then an optimal stochastic control problem, rather than just a study of the physical dynamics of the system. Shoemaker and co-workers [3] have solved similar problems, but for discrete-time, deterministic versions of the problem and using differential dynamic programming. In this paper, the problem is one of continuous-time stochastic dynamic programming.

## 2. Groundwater Management Application Model

The model considered here is a continuous time, stochastic general-

*This work has been supported in part by National Science Foundation's Computational Mathematics Program Grant DMS 93–1107, with supercomputer support from Los Alamos National Laboratory's Advanced Computing Laboratory, National Center for Supercomputing Applications, and Pittsburgh Supercomputer Center. This paper was published in *Proc. International Conference on Control and Information*, The Chinese University Press, Chinese University of Hong Kong, Sha Tin, Hong Kong, pp. 145-150, June 1995.

†This author's affiliation during the preparation of this paper for Fall 1994 was Division of Applied Mathematics, Brown University and for the Spring 1995 semester is School of Civil and Environmental Engineering, Cornell University.

ization of the discrete-time, deterministic optimal control model of Shoemaker and co-workers [3], a modification of a model of Ahlfeld et al. [1]. The science of groundwater models is covered in detail by Bear and Verruijt [2].

Some of the model assumptions are:

1. Contamination is rare stochastic event.
2. Detection is by monitor wells at selected nodes.
3. Remediation is by pumping out concentrations of pollution.
4. Pumping extraction is by wells, at selected nodes.
5. Groundwater is saturated in confined aquifer.

For dimension simplicity, the geometry of the confined three-dimensional aquifer is averaged over the vertical coordinate $z = \zeta$, letting $(x, y) = (\xi, \eta)$ be the horizontal coordinates and the bar ("—") denoting the vertical average. This is the *essentially horizontal aquifer* assumption [2]. The equation of continuity of mass when converted with the Boussinesq approximation to the vertically averaged hydraulic head $\overline{h}(\xi, \eta, t)$ becomes the flow transport dispersion equation [2, 1]

$$\begin{aligned} bS_s \frac{\partial \overline{h}}{\partial t} dt &= \left\{ \nabla_\xi \cdot (T \nabla_\xi \overline{h}) + f_a - \sum_{i \in \mathcal{P}} \overline{q}_i \delta(\xi - \xi_i, \eta - \eta_i) \right\} dt \\ &+ \quad bS_s H_c(t) \delta(\xi - \xi_{j_c}, \eta - \eta_{j_c}) dP(t), \end{aligned} \tag{1}$$

where $b$ is the aquifer thickness, $S_s$ is its storativity, $T = b \cdot K$ its the transmissivity, $\overline{q}_i$ is the extracting pump rate for the $i$th provided that $i$ is a pump node ($i \in \mathcal{P}$), and $f_a$ is the leakage flux from aquifers above or below. The pollution discharge as a rare stochastic event is modeled by the differential Poisson process $dP(t)$ at aquifer node $j_c$ with amplitude $H_c(t)$ (i.e., the head jumps at mean rate $\lambda \cdot H_c$ upon contamination). Here, $\nabla_\xi$ denotes the gradient with respect to the vector $\overline{\xi} = (\xi, \eta)$.

Flow though the porous media of the aquifer is very slow, so that the momentum balance is approximated by Darcy's law [2],

$$\overline{\mathbf{v}} = -K \nabla_\xi \overline{h}, \tag{2}$$

for Darcy's velocity $\overline{\mathbf{v}}$, where $K$ is the hydraulic conductivity tensor.

The contaminant transport equation for the vertically averaged contaminant concentration $\overline{c}(\xi, \eta, t)$ is the dispersion-diffusion equation [2, 1]

$$\begin{aligned} \theta b R_d \frac{\partial \overline{c}}{\partial t} dt &= \left\{ \nabla_\xi^T (b \widetilde{D} \nabla_\xi \overline{c}) dt - b \overline{\mathbf{v}}^T \nabla_\xi (\overline{c}) + f_a (c_a - \overline{c}) \right\} dt \\ &+ \quad \theta b R_d C_c(t) \delta(\xi - \xi_{j_c}, \eta - \eta_{j_c}) dP(t), \end{aligned} \tag{3}$$

where $\theta$ is the effective porosity of the aquifer, $R_d$ is the retardation coefficient, $f_a \cdot (c_a - \overline{c})$ is the contaminant leakage flux at concentration $c_a$, and $C_c(t)$ is the contaminant concentration amplitude (i.e., the jump in concentration at node $j_c$, with mean rate $\lambda \cdot C_c$).

Nonlinearity appears in two ways in the contaminant transport equation: first through the convection of the concentration at the Darcy velocity $-b\overline{\mathbf{v}}^T \cdot \nabla_\xi(\overline{c})$ and secondly through the hydrodynamic dispersion tensor, which includes both convective and molecular diffusion, so is velocity dependent:

$$\widetilde{D}(\overline{\mathbf{v}}) = d_M \begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix} + \overline{\mathbf{v}}^T \begin{bmatrix} D_{11} & D_{12} \\ D_{21} & D_{22} \end{bmatrix} \frac{\overline{\mathbf{v}}}{||\overline{\mathbf{v}}||}, \tag{4}$$

with diffusion coefficient $d_M$, dispersivity coefficients $D_{ij}$, and the magnitude of Darcy's velocity is given by: $||\overline{\mathbf{v}}|| = \sqrt{v_\xi^2 + v_\eta^2}$.

Note that the pump rate, $\overline{q}_i$, does not directly appear in contaminant transport equation, since pumping by extraction is at existing concentration levels, but extraction indirectly affects contaminant transport through the head and Darcy's velocity. The convection or drift term makes the contaminant transport equation bilinear in the unknown state vector

$$\overline{\mathbf{X}} \equiv \begin{bmatrix} \overline{h} \\ \overline{c} \end{bmatrix},$$

but the hydrodynamic tensor introduces higher order nonlinearities. Hence, the governing dynamics is a set of coupled, quasi-bilinear *diffusion* equations (1,3) with (2). The objective is to remediate the effects of the pollution contaminant once it is observed at monitor wells with the pump extraction rates as the control vector,

$$\overline{\mathbf{U}} = [\overline{\mathbf{U}}_i]_{n_p \times 1} \equiv \overline{I}_p \overline{q} = [\overline{q}_{i_p(i)}]_{n_p \times 1},$$

where $i_p(i)$ is the node number of the $i$th pump and $\overline{I}_p = [\delta_{i_p(i),j}]_{n_p \times \hat{n}}$ is the pump node indicator array, $n_p$ is the total number of pump nodes (control space dimension) and $\hat{n}$ is the total number of spatial nodes. Clearly, $n_p \le \hat{n}$.

The state dynamical systems can be summarized in the array-vector form:

$$\begin{aligned} \overline{E} \frac{\partial \overline{\mathbf{X}}}{\partial t} dt &= \mathcal{L}_\xi[\overline{\mathbf{X}}] dt \equiv \left\{ \overline{A}[\overline{\mathbf{X}}] + \overline{B} \cdot \overline{\mathbf{U}} \right\} dt \\ &+ \quad \overline{\mathbf{X}}_c(t) \delta(\overline{\xi} - \overline{\xi}_{j_c}) dP(t), \end{aligned} \tag{5}$$

where

$$\overline{\xi} \equiv \begin{bmatrix} \xi \\ \eta \end{bmatrix}, \quad \overline{E} = b \begin{bmatrix} S_s & 0 \\ 0 & \theta R_d \end{bmatrix},$$

the purely state dependent term is

$$\overline{A}[\overline{\mathbf{X}}] = \nabla_\xi \cdot (\overline{AL} \cdot \nabla_\xi \overline{\mathbf{X}}) + \overline{AN} \left[ \nabla_\xi \overline{\mathbf{X}} \right] + f_a \begin{bmatrix} 1 \\ c_a - \overline{c} \end{bmatrix},$$

the linear dispersion coefficient

$$\overline{AL} = \begin{bmatrix} T & 0 \\ 0 & 0 \end{bmatrix},$$

the nonlinear portion is

$$\overline{AN} \left[ \nabla_\xi \overline{\mathbf{X}} \right] = \begin{bmatrix} 0 \\ \nabla_{\overline{\xi}} (b \widetilde{D}(\overline{\mathbf{v}}) \nabla_{\overline{\xi}} \overline{c}) - b \overline{\mathbf{v}} \cdot \nabla_{\overline{\xi}} \overline{c} \end{bmatrix},$$

the purely linear control term is

$$\overline{B} = \begin{bmatrix} -\overline{\Delta}^T \cdot \overline{I}_p^T \\ [0]_{1 \times n_p} \end{bmatrix} \text{ with } \overline{\Delta} \equiv [\delta(\overline{\xi} - \overline{\xi}_i)]_{\hat{n} \times 1},$$

and the state contamination amplitude is

$$\overline{\mathbf{X}}_c = b \begin{bmatrix} S_s H_c \\ \theta R_d C_c \end{bmatrix}.$$

The vector state equation is completely specified, tentatively assuming the control vector is known, by the given vector state initial conditions,

$$\overline{\mathbf{X}} = \overline{\mathbf{X}}_0 = \begin{bmatrix} \overline{h}_0 \\ \overline{c}_0 \end{bmatrix} (\overline{\xi}) \text{ at } t = 0,$$

and general linear type vector state boundary conditions,

$$\overline{\alpha} \cdot (\overline{\mathbf{e}}_n \cdot \nabla_{\overline{\xi}}) \overline{\mathbf{X}} = \overline{\beta} \cdot (\overline{\mathbf{X}} - \overline{\mathbf{X}}_b) + \overline{\Gamma},$$

for all boundary points $\overline{\xi} = \overline{\xi}_b \equiv (\xi_b, \eta_b)$, where $\overline{\alpha}$, $\overline{\beta}$ and $\overline{\Gamma}$ are constant arrays, while $\overline{\mathbf{e}}_n$ is the unit normal to the boundary.

### 3. Spatial Finite Element Representation of Dynamics

Instead of moving directly to a stochastic dynamic programming formulation for distributed parameter control, the two-dimensional averaged aquifer geometry is converted to a spatial Galerkin finite element representation to save on the extra complexity of distributed parameter control. The horizontal aquifer is partitioned using $\hat{n} = n_\xi \times n_\eta$ into $(n_\xi - 1)\cdot(n_\eta - 1)$ rectangular elements, with $n_\xi$ and $n_\eta$ being the number of nodes in $\xi$ and $\eta$ directions, respectively. This partitioning is illustrated in Figure 1 for an example of a rectangular horizontal aquifer.

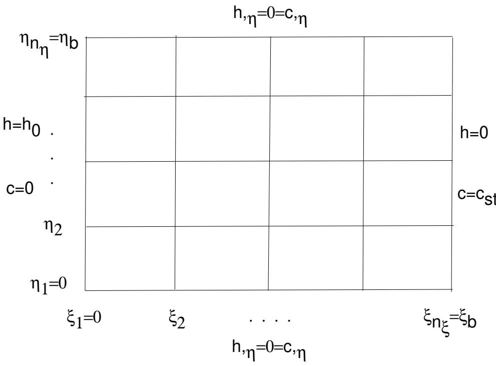


Figure 1: Spatial partitioning into elements for rectangular horizontal aquifer test case with sample boundary conditions. The general flow is from left to right with no flow at the bottom and top boundaries, while the state is specified on on the left and on the right, assuming standards have been met.

The spatial Galerkin Finite Element Method (FEM) approximation for the state vector is given by

$$\overline{\mathbf{X}}(\overline{\xi}, t) \approx \widetilde{\mathbf{X}}(\overline{\xi}, t) = \sum_{j=1}^{\hat{n}} \widehat{X}_j(t)\omega_j(\overline{\xi}), \tag{6}$$

where the spatial shape or basis functions, $\omega_i(\overline{\xi})$, for nodes $i = 1$ to $\hat{n}$, counting in row order, form a bilinear basis on the rectangular elements with the property that $\omega_j(\overline{\xi}_i) = \delta_{ij}$, forcing the state approximand to have the nodal state value: $\widetilde{\mathbf{X}}(\overline{\xi}_i, t) = \widehat{X}_i(t)$ at node $i$. The ("$\widehat{\ }$") notation denotes the spatial FEM approximation. Define the set $N_{nd}$ to be the set of all nodes, $i$, such that node $i$ is not a Dirichlet finite element node. The spatial node values of the state vector are determined by the Galerkin variational equation for weighted residuals of the state dynamic equation (5)

$$0 = \int_{\mathcal{D}_\xi} d\overline{\xi}\omega_i(\overline{\xi}) \left\{ \overline{E}\frac{\partial \widetilde{\mathbf{X}}}{\partial t} - \mathcal{L}_\xi[\widetilde{\mathbf{X}}] \right\}, \tag{7}$$

for all finite element nodes $i \in N_{nd}$ on horizontal aquifer domain $\mathcal{D}_\xi$ and subject to integration by parts to reduce all second order derivatives to first order. By formal expansion one can show that the local integration error is $O(||\Delta\overline{\xi}||^4)$ with respect to the weights for all nodes $i \in N_{nd}$ within the two-dimensional aquifer.

Upon substitution of the spatial Galerkin approximations for state, and a corresponding one for control, into the Galerkin equation (7) along with the usual derivative order reduction integration by parts (Green's Theorem), the resulting global spatial finite element equation for the spatial state vector $\widehat{\mathbf{X}}$ is the vector stochastic differential equation

$$\widehat{E}d\widehat{\mathbf{X}} = \left\{ \widehat{A}[\widehat{\mathbf{X}}] + \widehat{B}\widehat{\widehat{\mathbf{U}}} \right\} dt + \widetilde{\mathbf{X}}_c(t)dP(t), \tag{8}$$

with discrete state block vector

$$\widehat{\mathbf{X}}(t) = [\widehat{X}_i(t)]_{2\hat{n}\times 1} = \begin{bmatrix} [\hat{h}_i(t)]_{\hat{n}\times 1} \\ [\hat{c}_i(t)]_{\hat{n}\times 1} \end{bmatrix} = \begin{bmatrix} \hat{\mathbf{h}}(t) \\ \hat{\mathbf{c}}(t) \end{bmatrix},$$

(note that the Dirichlet nodes are excluded from $\widehat{\mathbf{X}}$), and $\widehat{\widehat{\mathbf{U}}}$ is a corresponding $\hat{n}\cdot n_p \times 1$ control vector. In (8), the lead coefficient is the

$$\widehat{E} = b\begin{bmatrix} S_s W & \vec{0} \\ \vec{0} & \theta R_d W \end{bmatrix},$$

which is based upon the pure FEM mass array

$$W = \left[W_{ij}\right]_{\hat{n}\times\hat{n}} = \left[\int_{\mathcal{D}_\xi} d\overline{\xi}\omega_i(\overline{\xi})\omega_j(\overline{\xi})\right]_{\hat{n}\times\hat{n}}$$

the inner product of the vector of basis functions with itself. The nonlinear state dependent array has the decomposition,

$$\begin{aligned} \widehat{A}[\widehat{\mathbf{X}}] &= f_a\begin{bmatrix} \mathbf{W}^{(1)} \\ c_a\mathbf{W}^{(1)} - W\hat{\mathbf{c}} \end{bmatrix} \\ &+ b\begin{bmatrix} KW^{(2)} & \vec{0} \\ \vec{0} & \widetilde{DW}^{(2)}(\widehat{\mathbf{X}}) \end{bmatrix}\widehat{\mathbf{X}} \\ &+ b\begin{bmatrix} \vec{0} \\ K\cdot \mathrm{Trace}_{23}[W^{(3)}\hat{\mathbf{h}}^T\hat{\mathbf{c}}] \end{bmatrix}, \end{aligned}$$

with FEM constant leakage term array

$$\mathbf{W}^{(1)} = \left[W_i^{(1)}\right]_{\hat{n}\times 1} = \left[\int_{\mathcal{D}_\xi} d\overline{\xi}\omega_i(\overline{\xi})\right]_{\hat{n}\times 1},$$

with FEM seepage and boundary array

$$\begin{aligned} W^{(2)} &= \left[-\int_{\mathcal{D}_\xi} d\overline{\xi}\nabla_\xi^T(\omega_i(\overline{\xi}))\cdot\nabla_\xi(\omega_j(\overline{\xi})) \right. \\ &+ \left.\int_{\partial\mathcal{D}_\xi} d\overline{\sigma}\omega_i(\overline{\xi})\cdot(\overline{\mathbf{e}}_n^T\cdot\nabla_\xi)\omega_j(\overline{\xi})\right]_{\hat{n}\times\hat{n}}, \end{aligned}$$

with FEM nonlinear dispersion and boundary array

$$\begin{aligned} \widetilde{DW}^{(2)} &= \left[-\int_{\mathcal{D}_\xi} d\overline{\xi}\widetilde{D}(\mathbf{V}_d)\nabla_\xi^T(\omega_i(\overline{\xi}))\nabla_\xi(\omega_j(\overline{\xi})) \right. \\ &+ \left.\int_{\partial\mathcal{D}_\xi} d\overline{\sigma}\widetilde{D}(\mathbf{V}_d)\omega_i(\overline{\xi})\cdot(\overline{\mathbf{e}}_n^T\nabla_\xi)\omega_j(\overline{\xi})\right]_{\hat{n}\times\hat{n}}, \end{aligned}$$

with the FEM approximation to the Darcy velocity given by $\mathbf{V}_d \equiv -K\sum_k \hat{h}_k\nabla_\xi\omega_k(\overline{\xi})$, and with FEM triadic Darcy convection array

$$W^{(3)} = \left[\int_{\mathcal{D}_\xi} d\overline{\xi}\omega_i(\overline{\xi})\nabla_\xi^T(\omega_j(\overline{\xi}))\nabla_\xi(\omega_k(\overline{\xi}))\right]_{\hat{n}\times\hat{n}\times\hat{n}}$$

using $\mathrm{Trace}_{23}[AB] = A : B = \sum_{jk} A_{ijk}B_{jk}$.

The control coefficient array is given by

$$\widehat{B} = [\widehat{B}_{i,j}]_{2\hat{n}\times\hat{n}\cdot n_p} = -\begin{bmatrix} \widehat{W}_p^{(2)} \\ \vec{0} \end{bmatrix},$$

$$\text{with } \widehat{W}_p^{(2)} = \left[\left[\delta_{i,i_p(k)}\delta_{j,i_p(k)}\right]_{\hat{n}\times\hat{n}}\right]_{1\times n_p}.$$

The stochastic pollution input array is

$$\widetilde{\mathbf{X}}_c(t) = [\widetilde{X}_{c,i}]_{2\hat{n}\times 1} = b\begin{bmatrix} S_s H_c(t)\mathbf{W}^{(c)} \\ \theta R_d C_c(t)\mathbf{W}^{(c)} \end{bmatrix},$$

using $\mathbf{W}^{(c)} = [W_i^c]_{\hat{n}\times 1} = [\omega_i(\overline{\xi}_c)]_{\hat{n}\times 1} = [\delta_{i,j_c}]_{\hat{n}\times 1}$.

Using the above notation the spatial SDE (8) can be renormalized as the spatial finite element SDE upon formally removing the mass array $\widehat{E}$ from the left hand side,

$$d\widehat{\mathbf{X}} = \widehat{E}^{-1}\left[\widehat{A}[\widehat{\mathbf{X}}] + \widehat{B}\cdot\widehat{\mathbf{U}}\right]dt + \widehat{\mathbf{X}}_c(t)dP(t). \tag{9}$$

with effective contaminant amplitude vector

$$\widehat{\mathbf{X}}_c(t) = \widehat{E}^{-1}\cdot\widetilde{\mathbf{X}}_c(t) = b\cdot\widehat{E}^{-1}\cdot\begin{bmatrix} S_s\cdot H_c\mathbf{W}^{(c)} \\ \theta R_d C_c\mathbf{W}^{(c)}\end{bmatrix}, \tag{10}$$

which is the scaled pollution amplitude to use in the Bellman equation.

## 4. Computational Stochastic Dynamic Programming

The optimal control problem is for the minimized treatment and pumping costs,

$$\overline{J}^*(\widehat{\mathbf{x}}, t) = \min_{\widehat{\mathbf{u}}}\left[\operatorname*{mean}_{P}\left[J[\widehat{\mathbf{X}},\widehat{\mathbf{U}},t]\,\Big|(\widehat{\mathbf{X}},\widehat{\mathbf{U}})(t) = (\widehat{\mathbf{x}},\widehat{\mathbf{u}})\right]\right] \tag{11}$$

where integral objective for minimal costs has the form

$$J[\widehat{\mathbf{X}},\widehat{\mathbf{U}},t] = \int_t^{t_f} dt\, C(\widehat{\mathbf{X}}(t),\widehat{\mathbf{U}}(t),t) \tag{12}$$

using instantaneous cost model:

$$\begin{aligned} C(\widehat{\mathbf{x}},\widehat{\mathbf{u}},t) \;&=\; e^{-\delta t}\left\{\sum_{i=1}^{n_p}\widehat{u}_i\left[a_{T,i} + a_{P,i}\Delta h_{P,i_p(i)}\right.\right. \\ &\left.\left. + \varepsilon_Q\widehat{u}_i\right] \;+\; \sum_{i\in\mathcal{M}} a_{M,i}\Delta c_{M,i}\right\} \end{aligned} \tag{13}$$

The cost parameters are as follows:

- $\delta$ is the real discount rate (corrected for inflation),
- $a_{T,i}$ is the contaminant treatment cost per unit pumping rate,
- $a_{P,i}$ is the pumping cost per unit pumping rate per unit lift,
- $a_{M,i}$ = is the cost of fines or their collection for violations of clean water standards for each node $i$ with violation and in $\mathcal{M}$,
- $\Delta h_{P,i_p(i)} = h^{\text{ground}}_{\text{level},i} - h^{\text{aquifer}}_{\text{bottom},i_p(i)}$ is the pump head lift,
- $\Delta c_{M,i} = c_i - C_{\text{std}}$ is the violation over standard at $i \in \mathcal{M}$,
- $\varepsilon_Q$ is the quadratic cost coefficient.

The optimization will have a number of domain constraints. The final pollution concentrations are constrained by $c_j(t_f) \leq C_{\text{std}}$ on observation wells at the final time, $t_f$, to ensure that final water contaminant concentrations satisfy government groundwater quality regulations. Similarly, within the time horizon, the state domain is generally constrained to be non-negative and finite by $0 \leq \hat{h}_i(t) \leq h_{\max,i}$ for the heads, and $0 \leq \hat{c}_i(t) \leq c_{\max,i}$ for the concentration. The controls or pumping rates are constrained by

$$0 \leq \widehat{U}_i(\widehat{\mathbf{x}}, t) \leq Q_{\max,i} \tag{14}$$

due to pumping extraction limits (non-negative too, since only extraction is allowed here) for all pumping nodes, $i = 1, n_p$.

Next, the *Principle of Optimality* is applied to this optimal stochastic control problem, leading to the *Partial Differential Equation of Stochastic Dynamic Programming*,

$$\begin{aligned} 0 \;&=\; \overline{J}^*_{,t}(\widehat{\mathbf{X}},t) + \mathcal{L}_x\left[\overline{J}^*\right](\widehat{\mathbf{X}},t) \\ &\equiv\; \overline{J}^*_{,t}(\widehat{\mathbf{X}},t) + \lambda\left[\overline{J}^*(\widehat{\mathbf{X}}+\widehat{\mathbf{X}}_c(t),t) - \overline{J}^*(\widehat{\mathbf{X}},t)\right] \\ &+\; S^*(\widehat{\mathbf{X}},t), \end{aligned} \tag{15}$$

for this problem, where $\overline{J}^*(\widehat{\mathbf{X}}, t)$ is the optimal, expected value of the objective cost and $\widehat{\mathbf{X}} = [\widehat{X}_i]_{2\hat{n}\times 1}$ is the state vector. The feedback control vector $\widehat{\mathbf{U}} = [\widehat{U}_i(\widehat{\mathbf{x}},t)]_{\hat{m}\times 1}$ is the output argument of the optimal control or switch term $S^*$,

$$S^*(\widehat{\mathbf{X}},t) = \min_{\widehat{\mathbf{u}}\in\mathcal{D}_u}\left[S(\widehat{\mathbf{X}},\widehat{\mathbf{u}},t)\right] \tag{16}$$

with switch argument or cost augmented by shadow cost:

$$S(\widehat{\mathbf{X}},\widehat{\mathbf{U}},t) = C(\widehat{\mathbf{X}},\widehat{\mathbf{U}},t) + \mathbf{F}^T(\widehat{\mathbf{X}},\widehat{\mathbf{U}},t)\nabla_x\overline{J}^*(\widehat{\mathbf{X}},t). \tag{17}$$

The switch term $S$ in (17) is a pseudo-Hamiltonian for the optimal control problem and its optimization can contribute nonsmoothness as the control switches to a constraint.

The Poisson pollution discharge shows up in (15) as the difference term $\widehat{\mathbf{X}}_c(t) = [\widehat{X}_{c,i}(t)]_{2\hat{n}\times 1}$, which is the vector of the spatial finite element modified Poisson noise amplitude coefficient and $\lambda$ is the rate of Poisson jumps.

The Bellman equation (15) with (16,17) is the prime target of our computational methods. Of particular interest are the larger state spaces, those with more than several dimensions.

Although the Bellman equation is a scalar valued equation, its solution simultaneously produces the optimal, expected cost and the optimal control vector. However, the equation is also a nonlinear, functional PDE, with optimization embedded in the switch term and a difference term following from the Poisson noise used to introduce the pollution. In order to systematically solve the Bellman equation, very reasonable assumptions will be made about the cost and drift functions. First, the instantaneous cost function from (13) is a quadratic function of the control,

$$C(\widehat{\mathbf{X}},\widehat{\mathbf{U}},t) \;=\; C_0(\widehat{\mathbf{X}},t) + \mathbf{C}_1^T(\widehat{\mathbf{X}},t)\widehat{\mathbf{U}} + \widehat{\mathbf{U}}^T C_2(t)\widehat{\mathbf{U}}. \tag{18}$$

This is motivated by the idea that greater costs may be accompanied by greater inefficiencies, thus $C_2$ is assumed to be non-negative definite. Second, the drift dynamics are assumed to be a linear function of the control,

$$\mathbf{F}^T(\widehat{\mathbf{X}},\widehat{\mathbf{U}},t) \equiv \widehat{E}^{-1}\left[\widehat{A}[\widehat{\mathbf{X}}] + \widehat{B}\widehat{\mathbf{U}}\right] = \mathbf{F}_0(\widehat{\mathbf{X}},t) + F_1(\widehat{\mathbf{X}},t)\widehat{\mathbf{U}}. \tag{19}$$

The assumptions on $C$ and $\mathbf{F}$ are like the usual linear quadratic (*LQ*) regulator assumptions, except that the *LQ* part pertains only to the control and not to both control and state. The state is allowed to be fully nonlinear, as it is in many applications. The assumptions of control-quadratic costs and control-linear dynamics (*Control-LQ*) is not very constricting, since they can easily be placed by the corresponding local Taylor approximations followed by successive iterations.

These *Control-LQ* assumptions (18-19) imply that the regular or unconstrained control can be formally computed from the local critical points of the switch function, $\nabla_u[S(\widehat{\mathbf{X}},\widehat{\mathbf{U}},t)] = \mathbf{0}$, yielding the continuous vector

$$\mathbf{U}_{\text{reg}}(\widehat{\mathbf{X}},t) = -\frac{1}{2}C_2^{-1}(t)\left[\mathbf{C}_1 + F_1^T\nabla_x\overline{J}^*\right](\widehat{\mathbf{X}},t), \tag{20}$$

for $n_p$ admissible pump nodes only. Next, applying the pumping extraction constraints (14) on the regular control in (20) yields the optimal feedback control,

$$U_i^*(\widehat{\mathbf{X}},t) = \left\{\begin{array}{cc} 0, & U_{\text{reg},i}(\widehat{\mathbf{X}},t) \leq 0 \\ U_{\text{reg},i}, & 0 \leq U_{\text{reg},i} \leq Q_{\max,i} \\ Q_{\max,i}, & Q_{\max,i} \leq U_{\text{reg},i}(\widehat{\mathbf{X}},t) \end{array}\right\}, \tag{21}$$

for all pump nodes, $i = 1, n_p$.

Note that since the switch term $S$ in (17) is linear in the optimal cost gradient $\nabla_x\overline{J}^*(\widehat{\mathbf{X}},t)$ and quadratic in the control $\widehat{\mathbf{U}}$ using the Control-LQ assumptions in (18-19), the regular control (20) is linear in $\nabla_x \overline{J}^*(\widehat{\mathbf{X}}, t)$ while the optimal control (21) is only piece-wise linear in $\nabla_x \overline{J}^*(\widehat{\mathbf{X}}, t)$ due to the constraints. Consequently, the optimal switch term $S^*$ in (16) and the Bellman equation in which it appears will be piece-wise quadratic in $\nabla_x \overline{J}^*(\widehat{\mathbf{X}}, t)$. This nonlinearity adds complexity to the analysis and computation.

## 5. State Finite Element Representation for Bellman Equation

In order to handle the state computational complexity of the Bellman equation (15), we again rely on the finite element method, but this time for the state. A state Galerkin approximation is assumed for optimal costs

$$\overline{J}^*(\widehat{\mathbf{X}}, t) \approx \widehat{V}(\widehat{\mathbf{X}}, t) \equiv \sum_{j=1}^{n_n} V_j(t)\phi_j(\widehat{\mathbf{X}}), \tag{22}$$

where the shape or basis functions, $\phi_i(\widehat{\mathbf{X}})$, form an $n_n$-dimensional multilinear basis for state hypercubes $\{0 \le \widehat{X}_i \le \widehat{X}_{\max,i}\}$, having the analogous form and properties as the bilinear spatial basis $\omega_i(\vec{\xi})$, such as $\phi_i(\widehat{\mathbf{X}}_j) = \delta_{i,j}$, where $\widehat{\mathbf{X}}_j$ is the state vector at the $j$th node. Here, $n_n = n_s^{2\hat{n}}$ is the number of state nodes and $n_s$ are the number of nodes per state (assumed to be the same for each state).

The temporal value of the state nodal costs $V_j(t)$ are determined by the variational Galerkin FEM form of Bellman equation:

$$0 = \int_{\mathcal{D}_x} d\widehat{\mathbf{X}}\phi_i(\widehat{\mathbf{X}}) \left[\widehat{V}_t(\widehat{\mathbf{X}}, t) + \mathcal{L}_x[\widehat{V}](\widehat{\mathbf{X}}, t)\right], \tag{23}$$

with the basis functions $\phi_i$ as residual weights, but only for non-Dirichlet state nodes when $i = 1$ to $n_n$ and subject to the usual integration by parts using Green's theorem.

Substitution of the state Galerkin approximation into the Galerkin-Bellman equation (23) for the optimal cost leads to a matrix equation after reduction in each term of (22). First, the time derivative term reduces to

$$\int_{\mathcal{D}_x} d\widehat{\mathbf{X}}\phi_i(\widehat{\mathbf{X}})\widehat{V}_t(\widehat{\mathbf{X}}, t) = \mathbf{\Phi} \cdot \mathbf{V}'(t), \tag{24}$$

where $\Phi_{i,j} \equiv \int_{\mathcal{D}_x} d\widehat{\mathbf{X}}\phi_i(\widehat{\mathbf{X}})\phi_j(\widehat{\mathbf{X}})$ and $\mathbf{V}(t) \equiv \left[V_i(t)\right]_{n_n \times 1}$.

Note that the state Galerkin approximation to the costs $\widehat{V}(\widehat{\mathbf{X}}, t)$ is related to the cost vector $\mathbf{V}(t)$ by $\widehat{V}(\widehat{\mathbf{X}}, t) = \vec{\phi}^T(\widehat{\mathbf{X}}) \cdot \mathbf{V}(t)$ with $\vec{\phi}(\widehat{\mathbf{X}}) = [\phi_i(\widehat{\mathbf{X}})]_{n_n \times 1}$. Next, the Poisson difference term with time-dependent *advance* $\widehat{\mathbf{X}}_c(t)$ reduces to

$$\begin{aligned} &\textstyle\int_{\mathcal{D}_x} d\widehat{\mathbf{X}}\phi_i(\widehat{\mathbf{X}})\lambda[\widehat{V}(\widehat{\mathbf{X}} + \widehat{\mathbf{X}}_c(t), t) - \widehat{V}(\widehat{\mathbf{X}}, t)] \\ &= \lambda \textstyle\sum_{j=1}^{n_n} V_j(t)\Delta\Phi_{i,j} = \lambda\Delta\mathbf{\Phi} \cdot \mathbf{V}(t), \end{aligned} \tag{25}$$

with $\Delta\Phi_{i,j}(t) \equiv \int_{\mathcal{D}_x} d\widehat{\mathbf{X}}\phi_i(\widehat{\mathbf{X}})[\phi_j(\widehat{\mathbf{X}} + \widehat{\mathbf{X}}_c(t)) - \phi_j(\widehat{\mathbf{X}})]$.

The more complicated switch term takes more effort to reduce due to the optimization term. Starting from

$$\begin{aligned} \int_{\mathcal{D}_x} d\widehat{\mathbf{X}}\phi_i(\widehat{\mathbf{X}})S^*(\widehat{\mathbf{X}}, t) &= \int_{\mathcal{D}_x} d\widehat{\mathbf{X}}\phi_i(\widehat{\mathbf{X}}) \min_{\widehat{\mathbf{U}}}[C_0 + \mathbf{C}_1^T\widehat{\mathbf{U}} \\ +\widehat{\mathbf{U}}^T C_2\widehat{\mathbf{U}} &+ (F_0 + F_1\widehat{\mathbf{U}})^T \sum_j V_j(t)\nabla_x\phi_j(\widehat{\mathbf{X}})], \end{aligned} \tag{26}$$

from which is derived the regular control:

$$\widehat{\mathbf{U}}_{\text{reg}} = -\frac{1}{2}C_2^{-1}\left\{\mathbf{C}_1 + [(F_1^T\nabla_x)\vec{\phi}(\widehat{\mathbf{X}})]^T \cdot \mathbf{V}(t)\right\},$$

and subsequently the optimal control:

$$\widehat{\mathbf{U}}_i^*(\widehat{\mathbf{X}}, t) = \left\{ \begin{array}{cc} 0, & \widehat{\mathbf{U}}_{\text{reg},i}(\widehat{\mathbf{X}}, t) \le 0 \\ \widehat{\mathbf{U}}_{\text{reg},i}, & 0 \le \widehat{\mathbf{U}}_{\text{reg},i}(\widehat{\mathbf{X}}, t) \le Q_{\max,i} \\ Q_{\max,i}, & Q_{\max,i} \le \widehat{\mathbf{U}}_{\text{reg},i}(\widehat{\mathbf{X}}, t) \end{array} \right\},$$

which has the same form as in (21) before the state FEM representation. Summarizing, the quadrature ready switch term can be written as

$$\begin{aligned} \int_{\mathcal{D}_x} d\widehat{\mathbf{X}}\phi_i(\widehat{\mathbf{X}})S^*(\widehat{\mathbf{X}}, t) &= \int_{\mathcal{D}_x} d\widehat{\mathbf{X}}\phi_i(\widehat{\mathbf{X}}) \Big[C_0 + \mathbf{C}_1^T\widehat{\mathbf{U}}^* \\ +\widehat{\mathbf{U}}^{*T} C_2\widehat{\mathbf{U}}^* &+ [(F_0 + F_1\widehat{\mathbf{U}}^*)^T \nabla_x\vec{\phi}(\widehat{\mathbf{X}})]^T\mathbf{V}(t)\Big] \\ &\equiv \widehat{S}_i(\widehat{\mathbf{X}}, t) \end{aligned} \tag{27}$$

and the Galerkin-Bellman equation as

$$0 = \mathbf{\Phi} \cdot \mathbf{V}'(t) + \lambda\Delta\mathbf{\Phi}(t) \cdot \mathbf{V}(t) + \widehat{\mathbf{S}}(\widehat{\mathbf{X}}, t), \tag{28}$$

where the switch term $\widehat{\mathbf{S}}(\widehat{\mathbf{X}}, t)$ will need high order numerical quadrature. The multilinear product basis over hypercube (hyper-rectangular) elements should lead to local state space integral errors on the order of $O(||\Delta x||^{\hat{n}+2})$ for the $\hat{n}$-dimensional state space.

The prime difficulty, even considering massively parallel implementation, is the large total number of combined spatial-state hypercube generated nodes:

$$n_n = n_s^{2\hat{n}} = e^{2\hat{n}\log(n_s)}, \tag{29}$$

where $\hat{n} = n_\xi \times n_\eta$ is the product of the number of points in both the $\xi$-direction and $\eta$-direction. It is assumed that all the state vectors $\widehat{\mathbf{X}}_i$, for state nodes $i = 1$ to $n_n$, have the same number of nodes $n_s$. Hence, although state vector variable $\widehat{\mathbf{X}}$ is of dimension $n_x = 2\hat{n} \cdot n_s$ as a vector data structure, the dependence of the value $\widehat{V}(\widehat{\mathbf{X}}, t)$ on the $2\hat{n}$-dimensional vector $\widehat{\mathbf{X}}$ requires a $2\hat{n}$-dimensional hypercube data structure representation

$$\hat{\mathbf{\imath}} \equiv [\hat{\imath}_i]_{2\hat{n}\times 1} = (\hat{\imath}_1, \hat{\imath}_2, \ldots, \hat{\imath}_{2\hat{n}}) \tag{30}$$

where each current state node index runs from $\hat{\imath}_i = 1$ to $n_s$ leading to the $n_n = n_s^{2\hat{n}}$ possibilities in (29) when all state node indices are considered. Similarly, the control vector $\widehat{\mathbf{U}}(\widehat{\mathbf{X}}, t)$ requires a $2\hat{n} \cdot n_p$-dimensional hypercube data structure representation considering its $n_p$ control components. Thus, the exponential order of the total number of nodes is evidence of *Bellman's curse of dimensionality* for computational dynamic programming.

With hypercubic elements, the number of elements would be $n_e = (n_s - 1)^{2\hat{n}}$ with $n_{en} = 2^{2\hat{n}}$ nodes per element. Instead of the above hypercube data structure (30), it is better for coding, code optimization and load balancing [8] to use a vector data structure global node index,

$$\bar{\imath} = \bar{\imath}(\hat{\mathbf{\imath}}) = \bar{\imath}(\hat{\imath}_1, \hat{\imath}_2, \ldots, \hat{\imath}_{2\hat{n}}) \tag{31}$$

where $\bar{\imath} = 1$ to $n_n$ in order to include all $n_n$ hypercube node possibilities. However, the transformations between hypercube and vector data structures and their inverses is necessary when computing boundary conditions and gradients. A hypercube to vector data structure node index transformation is of the form:

$$\bar{\imath} = \bar{\imath}(\hat{\mathbf{\imath}}) = 1 + \sum_{j=1}^{2\hat{n}}(\hat{\imath}_j - 1)n_s^{(j-1)}, \tag{32}$$

motivated by Fortran-like representation with common size $n_s$. The vector to hypercube data structure node index inverse transformation has the form:

$$\hat{\imath}_j = 1 + \text{Int}\left[\frac{(\bar{\imath} - 1) - \sum_{k=j+1}^{2\hat{n}}(\hat{\imath}_k(\bar{\imath}) - 1) \cdot n_s^{(k-1)}}{n_s^{(j-1)}}\right], \tag{33}$$

for $j = 2\hat{n}$ to 1 step $-1$, depending on Fortran-like representation with common size $n_s$ and on the identity $\sum_{k=M+1}^{M} a_k \equiv 0$. These data structures are also helpful in constructing formulas for the multilinear state basis.

Mathur and Johnsson [11] have given a modified data parallel algorithm for the finite element method on the Connection Machine CM-5 using virtual processors for unassembled nodal points.

In order to handle the Bellman equation nonlinearities, an extrapolated, predictor-corrector, Crank-Nicholson method has been applied along with the finite difference method [9]. The application of finite element method to the stochastic optimal control problem has been examined in lower dimensions such as $\hat{n} = 1$ and $\hat{n} = 2$ on NCSA Cray-2 supercomputing system [4]. A general, robust technique for time-space mesh ratio selection gives promise of robust corrector convergence [14]. Current results indicate that the finite element method is stable enough for the hyperbolic form (when the Gaussian noise is zero). The finite element method requires fewer mesh points than the finite difference method, permitting the solution of larger problems [4].

In stochastic optimal control problems, most computations can be classified into the operations between large scale arrays of same size and operations between one different arrays array, the latter causing inefficient computational and memory movement. A decomposition technique and broadcasting technique help to modify these miss-matched array operations, so that they are executed only with the same size arrays, making them more efficient for massively parallel computations. In the finite difference method, we have attained good performance in both timing and storage requirements by using these techniques. We also expect to obtain the good results in the finite element applications.

Shoemaker and co-workers [13, 3] have parallelized discrete-time *differential dynamic programming* problems. Kushner and co-workers [12] use the *Markov chain approximation*, where the finite difference equations are constructed to preserve underling probability laws.

## 6. Massively Parallel Computational Results

The authors plan to have results on the new generation Connection Machine CM-5 or Cray C90 for the presentation of this paper, when preliminary tests of CM-5 or Cray C90 code conversions have been made. The stochastic programming code is well developed for resource models on the earlier Connection Machines, CM-2 and CM-200 [16, 17], as well as the earlier Crays.

## 7. Conclusions

In summary, the model for parallel computation of large scale groundwater remediation models has been developed as a stochastic optimal control problem in continuous time. The stochasticity enters through the random dumping of pollution into an aquifer modeled by the rare events of a Poisson process. The computational target is Bellman's PDE of stochastic dynamic programming. Galerkin finite element methods are used for both spatial and state variables. In addition, predictor-corrector methods are used to the nonlinear Galerkin-Bellman equation and mesh selection methods allow robust corrector convergence. These methods should prove useful for large scale groundwater remediation applications, and subsequently as a test for other *grand and national challenge* applications.

## 8. References


[1] D. P. Ahlfeld, J. M. Mulvey, G. F. Pinder and E. F. Wood, "Contaminated groundwater remediation design using simulation, optimization, and sensitivity theory: 1. Model development," *Water Resources Research*, vol. 24(3), pp. 431-441, Mar. 1988.

[2] J. Bear and A. Verruijt, *Modeling Groundwater Flow and Pollution*, Kluwer Academic Publishers, Boston, 1987.

[3] L.-C. Chang, C. A. Shoemaker, and P. L.-F. Liu, "Optimal time-varying pumping rates for groundwater remediation: Application of a constrained optimal control algorithm," *Water Resources Research*, vol. 28(12), pp. 3157-3173, Dec. 1992.

[4] S.-L. Chung, F. B. Hanson and H. H. Xu, "Parallel stochastic dynamic programming: Finite element methods," *Lin. Alg. Applic.*, vol. 172, pp. 197-218, Jul. 1992.

[5] J. Dantas de Melo, J. L. Calvet and J. M. Garcia, "Vectorization and multitasking of dynamic programming in control: Experiments on a Cray-2," *Parallel Computing*, vol. 13, pp. 261-269, 1990.

[6] *Future Directions in Control Theory: A Mathematical Perspective*, W. H. Fleming, Chairman, Society for Industrial and Applied Mathematics, Philadelphia, 1988.

[7] *Grand Challenges: High Performance Computing and Communications*, Committee on Physical, Mathematical, and Engineering Sciences, Federal Coordinating Council for Science, Engineering, and Technology, Washington, DC, 1991.

[8] F. B. Hanson, "Stochastic dynamic programming: Advanced computing constructs," *Proc. 28th IEEE Conf. on Decision and Control, vol. I*, pp. 901-903, Dec. 1989.

[9] F. B. Hanson, "Computational dynamic programming on a vector multiprocessor," *IEEE Trans. Automatic Control 36(4)*, pp. 507-511, Apr. 1991.

[10] D. H. Jacobson and D. Q. Mayne, *Differential Dynamic Programming*, American Elsevier, New York, 1970.

[11] Z. Johan, K. K. Mathur, S. L. Johnsson, "An efficient communication strategy for finite element methods on the connection machine CM-5 system," *TMC Technical Report No. 256*, Aug. 6, 1993.

[12] H. J. Kushner and P. G. Dupuis, *Numerical Methods for Stochastic Control in Continuous Time*, Springer-Verlag, New York, 1992.

[13] L.-Z. Liao and C. A. Shoemaker, "Convergence in unconstrained discrete-time differential dynamic programming," *IEEE Trans. Automatic Control*, vol. 36(6), pp. 692-706, Jun. 1991.

[14] K. Naimipour and F. B. Hanson, "Convergence of a numerical method for the Bellman equation of stochastic optimal control with quadratic costs and constrained control," *Dynamics and Control*, vol. 3(3), pp.237-259, Jul. 1993.

[15] G. F. Pinder and E. O. Frind, "Application of Galerkin's procedures to aquifer analysis," *Water Resources Research*, vol. 8, pp. 108-120, Feb. 1972.

[16] H. H. Xu, F. Hanson and S. L. Chung, "Optimal data parallel methods for the stochastic dynamical programming," in *Proc. 1991 Int. Conf. Parallel Processing, vol. III: Algorithms & Applications*, pp. 142-146, Aug. 1991.

[17] H. H. Xu, F. Hanson and S. L. Chung, "Parallel solutions of dimensionality problems for the stochastic dynamical programming," in *Proc. 30th IEEE Conf. Decision and Control*, vol. 2, pp. 1717-1723, Dec. 1991.

[18] S. Yakowitz, "Dynamic programming applications in water resources," *Water Resources Research*, vol 18, pp. 673-696, 1982.