ELSEVIER

Physica D 154 (2001) 15–25

PHYSICA D

www.elsevier.com/locate/physd

Variational optimisation by the solution of a series of Hamilton–Jacobi equations

Prasana K. Venkatesh [a,*], Linda R. Petzold [b], Robert W. Carr [c], Morrel H. Cohen [d], Anthony M. Dean [d]

[a] *Schlumberger-Doll Research, Old Quarry Road, Ridgefield, CT 06877, USA*
[b] *Departments of Mechanical Engineering and Computer Science, University of California, Santa Barbara, CA 93106, USA*
[c] *Department of Chemical Engineering and Materials Science, University of Minnesota, Minneapolis, MN 55455, USA*
[d] *Corporate Research Laboratories, ExxonMobil Research and Engineering Company, Annandale, NJ 08801, USA*

Received 19 July 2000; received in revised form 29 December 2000; accepted 9 January 2001

Communicated by F.H. Busse

Abstract

We present a unified scheme which, by combining backward differentiation formulae and variational optimisation theory, significantly extends the scope of both concepts and their application to optimisation problems wherein the constituent dynamics consist of widely disparate scales. Our approach is based on the time-integration of a series of matrix Riccati expressions which satisfy a Hamilton–Jacobi equation. © 2001 Elsevier Science B.V. All rights reserved.

Keywords: Variational optimisation theory; Hamilton–Jacobi equation; Time-integration; Optimisation problem

1. Introduction

Three of the most significant developments in fluid dynamics over the past two decades have been the experimental discovery of coherent structures in certain flows [1], the introduction of the Karuhnen–Loeve decomposition, also known as proper orthogonal decomposition [2,3], for their resolution, and the attempt to control flows actively for engineering utility by means of manipulating these structures [4–8]. It is with the aspect of determining optimum control functions pertaining to fluid dynamical situations that we concern ourselves in this paper. It is considerably recognised that there are complicated and important problems of optimum control in continuous-time such as the control of Navier–Stokes turbulence [9–12] for which efficient computational means are necessary. There are also the problems of control of hydrodynamic interactions in a porous medium [13] which are by no means computationally simple.

In the recent past, dynamic programming [14–16], an important deviation from the classical treatment of Hamilton's principle in relation to the derivation of Lagrange's equations, has been of particular interest in the

* Corresponding author. Fax: +1-203-438-3819. Most of this work was done while affiliated with the Department of Chemical Engineering at the University of Minnesota.
E-mail addresses: prasana@slb.com, prasana@ridgefield.sdr.slb.com (P.K. Venkatesh).

PII: S0167-2789(01)00222-6

stochastic variational derivation of quantum mechanics [17–21]. It has also been recently shown [16] that for simple, time-dependent, discrete processes, dynamic programming is the efficient manner of computing optimum control functions [22]. In this paper, we point out that the classical theory of dynamic programming in combination with the theory of stiff differential equations [23–26] is of immense utility in computing optimum control functions of fluid dynamical systems whose dynamics possess disparate scales.

Consider an optimum control problem of minimising a functional $J(\phi)$ governed by the dynamics

$$\frac{\partial u}{\partial t} + \nu A u + R(u, \phi) = 0. \tag{1}$$

In Eq. (1), $u(x, t) \in R^d$ denotes the velocity field with t being the time and the vector x signifying the spatial coordinate. $\nu > 0$ is the kinematic viscosity and A denotes the *Stokesian* operator; R signifies the inertial and boundary terms and depends on the mass flux. ϕ is the control function which establishes the state u. The functional $J(\phi)$ may be taken to be convex in ϕ. A specific example of flow-dynamics represented by Eq. (1) is turbulent flow through a channel along with the no-slip boundary condition,

$$\frac{\partial u_i}{\partial t} + u_j \frac{\partial u_i}{\partial x_j} = -\frac{\partial p}{\partial x_i} + \frac{1}{Re} \frac{\partial^2 u_i}{\partial x_j \partial x_j}, \tag{2}$$

$$\frac{\partial u_j}{\partial x_j} = 0 \tag{3}$$

with

$$u_i|_w = \phi(t)\delta_{i2}. \tag{4}$$

Here, p is the pressure, Re the Reynolds number, u_i are the velocity components and the control ϕ is the wall-normal velocity at the wall-surface. The objective functional, $J(\phi)$, to be minimised may be taken to represent the wall-drag in turbulent flow. Another example, albeit more complex but in a different manner due to the presence of additional fluid phases, is the control of specific fluid mobility in a porous medium given by the following momentum and mass conservation equations:

$$u_i = -\frac{k k_{ri}(S_i)}{\mu_i}(\nabla p_i - \rho_i g), \tag{5}$$

and

$$\frac{\partial}{\partial t}(m\rho_i S_i) + \nabla \rho_i u_i = \phi(x, t). \tag{6}$$

For some specified initial and boundary conditions of the above dynamics one may seek, for example, to maximise the objective functional $-J(\phi)$ representing an oil-recovery utility function for a given reservoir dynamics [13]. In Eqs. (5) and (6), m denotes the porosity of the porous medium and k its permeability; the μ_i, u_i, S_i and ρ_i, are, respectively, the phase-viscosities (molecular), phase-velocities, phase-saturations, and phase-densities; the $k_{ri}(S_i)$ are the relative permeability functions specific to each phase and the control function ϕ denotes the displacement stimulus used to displace the desired subterranean fluid. Such optimum control problems arise in the primary and secondary recovery of oil and natural gas in reservoirs.

The method of determining optimum control functions described herein is general irrespective of the specific nature of the underlying fluid dynamics. Therefore, in the following we base our description of the method with respect to the formalism represented by Eq. (1). The essential idea in the determination of optimal control functions is the computation of a sequence of perturbations $\delta\phi$ to the control function ϕ using the sequence

$$\nabla_{\phi\phi} J \delta\phi = -\nabla_\phi J \quad \text{and} \quad \phi \leftarrow \phi + \delta\phi, \tag{7}$$

which is the method due to Newton. The computation of a sequence of Newton steps requires the full spatial and temporal resolution of the dynamics of Eq. (1) for every value of ϕ in the sequence. This is computationally very prohibitive for important cases of fluid flows like turbulent flows and flows through porous media. For the case of homogeneous isotropic turbulence the cost of a single simulation of Eq. (1) is greater than $Re^{8.75}$ where Re is the Reynolds number. Thus, due to such computational constraints, one has to be content with using a gradient approach to determining the optimum controls by means of the descent steps $\delta\phi \leftarrow -\nabla_\phi J$, which unfortunately possesses poor convergence properties. It is the purpose of this letter to show that the Newton step of Eq. (7) may be calculated efficiently for difficult hydrodynamic optimisation problems of the kind described by Eq. (1) by solving a series of Hamilton–Jacobi equations drawn from a dynamic programming formalism. The computational complexity per step of this method is $O([\Delta t]^{-1})$ whereas the computational complexity per step of the classical-Newton scheme is $O([\Delta t]^{-2})$ with Δt being the mesh size in the temporal integration of (1).

2. The problem of second variations

We begin by noting that the Newton–Raphson step of Eq. (7) is theoretically equivalent to computing the second variation, $\Delta^2 J(\phi)$, of the functional $J(\phi)$. Prior to computing this second variation, we expand the velocity field spectrally

$$u(x,t) \approx \sum_{i=0}^{n-1} w_i(t)\Psi_i(x) \tag{8}$$

in order to minimise the residual

$$\int_x \left\{ \left(\frac{\partial}{\partial t} + \nu A \right) u + R(u,\phi) \right\} \omega(x)\,\mathrm{d}x, \tag{9}$$

where we note that the $\{\Psi_i(x)\}$ form an orthonormal basis, n is the order of the first neglected basis function in the truncation of (8), and $\omega(x)$ takes the value of infinity at the roots of the first neglected basis function $\Psi_n(x)$ but is zero otherwise. The net result of the spectral transformation of Eqs. (8) and (9) is that we now seek to minimise the functional $J(\phi)$ subject to the dynamics

$$\frac{\mathrm{d}}{\mathrm{d}t} W(t) + f(W,\phi) = 0, \tag{10}$$

where $W(t) \in R^{dn}$. Without any loss of generality we shall assume that the cost functional $J(\phi)$ takes the form

$$J(\phi) = \Lambda(W(T_f), T_f) + \int_0^{T_f} L(W(t),\phi)\,\mathrm{d}t, \tag{11}$$

where Λ are L functionals which are also implicitly dependent on the state u and the control ϕ; T_f is some time duration over which control is sought.

The second variation $\Delta^2 J(\phi)$ is equivalent to another optimum control problem, namely, that of minimising a functional, in terms of perturbations δW and $\delta\phi$, defined by

$$\delta^{\mathrm{T}} W(T_f) \Lambda_{WW}(T_f) \delta W(T_f) + \frac{1}{2}\int_0^{T_f} [\delta^{\mathrm{T}} W \quad \delta^{\mathrm{T}}\phi] \begin{bmatrix} L_{WW} & L_{W\phi} \\ L_{\phi W} & L_{\phi\phi} \end{bmatrix}\Bigg|_{W^*,\phi^*} \begin{bmatrix} \delta W \\ \delta\phi \end{bmatrix} \mathrm{d}t \tag{12}$$

and which is constrained by the dynamics

$$\delta\dot{W} + \frac{\partial}{\partial W} f(W,\phi)\Big|_{W^*,\phi^*} \delta W + \frac{\partial}{\partial \phi} f(W,\phi)\Big|_{W^*,\phi^*} \delta\phi = 0. \tag{13}$$

We note that the superscript T in the equations stands for the adjoint operator and that the optimum trajectory is defined by W^* and ϕ^*. The perturbations, $\delta W(t) = W(t) - W^*(t)$ and $\delta\phi = \phi - \phi^*$, are defined with respect to the optimum trajectory. At the initial-time the perturbation in the state is known. The solution to the second variation is given by the Hamilton–Jacobi equation

$$-\frac{\partial}{\partial t} V^*(\delta W^*(t), t) = \text{infimum}\ \ H(\delta W^*(t), \delta\phi, t), \tag{14}$$

wherein the adjoint function $V^*(\delta W^*(t), t)$ is defined by

$$\begin{aligned} V^*(\delta W^*(t), t) = {} & \delta^{\mathrm{T}} W^*(T_f) \Lambda_{W^*W^*}(T_f) \delta W^*(T_f) \\ & + \frac{1}{2}\int_t^{T_f} [\delta^{\mathrm{T}} W^* \quad \delta^{\mathrm{T}}\phi^*] \begin{bmatrix} L_{WW} & L_{W\phi} \\ L_{\phi W} & L_{\phi\phi} \end{bmatrix}\Bigg|_{W^*,\phi^*} \begin{bmatrix} \delta W^* \\ \delta\phi^* \end{bmatrix} \mathrm{d}t \end{aligned} \tag{15}$$

and the Hamiltonian $H(\delta W^*(t), \delta\phi, t)$ is defined as

$$H(\delta W^*(t), \delta\phi, t) = [\delta^{\mathrm{T}} W^* \quad \delta^{\mathrm{T}}\phi] \begin{bmatrix} L_{WW} & L_{W\phi} \\ L_{\phi W} & L_{\phi\phi} \end{bmatrix}\Bigg|_{W^*,\phi^*} \begin{bmatrix} \delta W^* \\ \delta\phi \end{bmatrix} - \delta^{\mathrm{T}}\lambda\delta\dot{W}^*. \tag{16}$$

Note that $\delta\lambda$ in Eq. (16) is a vector of Lagrange multipliers. The solution of a general Hamilton–Jacobi equation is usually determined to be the zero-viscosity level set solution of the propagating interfaces [27] and this is a complicated process. But fortunately, the Hamilton–Jacobi equation (14) is of a conducive structure and its solution is analytically given by the feedback law

$$\delta\phi = L^{-1}_{\phi^*\phi^*}(t) \left[\mathfrak{R}^{\mathrm{T}}(t) \left(\frac{\partial}{\partial\phi} f(W,\phi)\Big|_{W^*,\phi^*} \right) - L_{W^*\phi^*}(t) \right]^{\mathrm{T}} \delta W(t), \tag{17}$$

where $\mathfrak{R}(t)$ is the solution to the matrix Riccati equation

$$\begin{aligned} \dot{\mathfrak{R}}(t) = {} & \left[\left(\frac{\partial}{\partial W} f(W,\phi)\Big|_{W^*,\phi^*} \right) - \left(\frac{\partial}{\partial\phi} f(W,\phi)\Big|_{W^*,\phi^*} \right) L^{-1}_{\phi^*\phi^*}(t) L_{\phi^*W^*}(t) \right] \mathfrak{R}^{\mathrm{T}}(t) \\ & + \mathfrak{R}(t) \left[\left(\frac{\partial}{\partial W} f(W,\phi)\Big|_{W^*,\phi^*} \right) - \left(\frac{\partial}{\partial\phi} f(W,\phi)\Big|_{W^*,\phi^*} \right) L^{-1}_{\phi^*\phi^*}(t) L_{\phi^*W^*}(t) \right] \\ & + \mathfrak{R}(t) \left(\frac{\partial}{\partial\phi} f(W,\phi)\Big|_{W^*,\phi^*} \right) L^{-1}_{\phi^*\phi^*}(t) \left(\frac{\partial}{\partial\phi} f(W,\phi)\Big|_{W^*,\phi^*} \right)^{\mathrm{T}} \mathfrak{R}(t) - L_{W^*W^*}(t) \\ & + L_{W^*\phi^*}(t) L^{-1}_{\phi^*\phi^*}(t) L_{\phi^*W^*}(t) \end{aligned} \tag{18}$$

and obeys the terminal condition

$$\mathfrak{R}(T_f) = \Lambda_{W^*W^*}(T_f). \tag{19}$$

This now establishes the methodology for computing the perturbations to the control ϕ in the Newton method defined by Eq. (7). The only remaining aspect is the integration of the matrix Riccati equation which is essentially an initial value problem of the form

$$\dot{\mathfrak{R}}(t) = F(\mathfrak{R}(t)) \tag{20}$$

after making a change of sign and a translation along the temporal axis. The matrix Riccati equation will be a stiff differential equation due to the existence of disparate time scales and is therefore best integrated using backward

differentiation formulae (BDF). There are many such formulae [23–26]; for instance, a K-step formulae consists of replacing all time derivatives of the Riccati matrix by the derivative of the polynomial which interpolates the computed solution at times $t_i, t_{i-1}, \ldots, t_{i-K}$ where t_i is the current time-step in the solution process. This requires the method of Newton–Raphson to find the roots of

$$F\left(\mathfrak{R}_i, t_i, \frac{\sum_{l=0}^{K}\alpha_l \mathfrak{R}_{i-l}}{t_{i+1}-t_i}\right) = 0, \tag{21}$$

where the $\{\alpha_l\}$ are the coefficients of the BDF scheme and we remind the reader that the unknown in Eq. (21) is the Riccati matrix at the instant of time t_i. The numerical stability and the accuracy of any BDF is dependent to a great extent on these coefficients. In the example problem considered later in this paper, we choose a K step implicit BDF with a predictor–corrector scheme [25]. We allow K and the step size to vary in our implementation depending on the dynamics and stiffness. Thus, the total number of time-steps for each matrix Riccati differential equation is determined by the BDF. BDF have a fixed computational cost irrespective of the eigenspectrum of the linearised part of the right-hand side of Eq. (20) over the entire time-horizon. This now completes the entire description of the scheme for computing optimum control functions. It should be stressed that the dynamic programming formalism presented here possesses the property of quadratic convergence. We have found that it is more efficient than the conventionally used gradient method of computing optimum controls [5,22] which does not make any use of curvature information at all. Additionally, it is also more efficient than the classical manner of directly computing the Newton steps given by Eq. (7).

3. An illustration: control of two-phase porous medium flow

In order to illustrate how our method works in practice, we present results for the optimum control of a time-dependent, two-dimensional, incompressible, two-phase flow in a porous medium over a square spatial domain [13] in the absence of capillary forces. The problem concerns the flow of an oleic phase and an aqueous phase with one point-source and one point-sink, respectively located, at diagonally opposite ends of the square domain. Formulating in terms of the aqueous-phase saturation, S, we have

$$\frac{\partial S}{\partial t} + \nabla\cdot(\vec{u}\xi(S)) = 0, \tag{22}$$

$$\frac{\partial S}{\partial n} = 0 \quad \text{on } \partial\Omega, \tag{23}$$

$$\vec{u} = -\lambda(S)\nabla p \quad \text{and} \quad \nabla\cdot(\lambda(S)\nabla p) = g, \tag{24}$$

$$S(t,0,0) = 1 \quad \text{and} \quad \frac{\partial p}{\partial n} = 0 \quad \text{on } \partial\Omega. \tag{25}$$

$\Omega = \{(x,y)|0 \le x \le 1, 0 \le y \le 1\}$ is the domain of the medium and $\partial\Omega$ its boundary; $\vec{u}$ the Darcy total phase-velocity vector, p the pressure, and g the acceleration due to gravity. The point-source is located at the origin $(0,0)$ and the point-sink is located at the coordinates $(1,1)$. The aqueous fractional flow curve $0 \le \xi(S) \le 1$ is taken to be

$$\xi(S) = \frac{S^2}{\lambda(S)}, \tag{26}$$

where $\lambda(S)$ is the total mobility function, i.e., the ratio of the relative permeability to the viscosity. The optimal control problem, albeit a very theoretical one, is the determination of the optimum total mobility function $\lambda(S)$, i.e.,

$\phi \equiv \lambda(S)$, in order to maximise the average oleic phase constituent at the point-sink over the time-domain T_f of control. Mathematically, the objective is to minimise

$$J(\phi) = \frac{1}{T_f}\int_0^{T_f} S(t, 1, 1)\,\mathrm{d}t. \tag{27}$$

Note that, at the initial-time, the phase-saturation at the point-source is wholly aqueous and we require that $0 \leq S(t, x, y) \leq 1$ always be satisfied. We specify the time-horizon, T_f, of the control-system, to also be an optimisable variable. The objective functional strives to delay the onset of viscous fingering and prior to the onset, it is to be desired that as much of the oleic phase is displaced by the aqueous phase towards the point-sink.

Our aim is to illustrate the efficiency of dynamic programming over the classical method of adjoint states and gradients [22] in the computation of the optimum control. We also present comparisons of dynamic programming with the classical manner of directly computing Newton steps by Eq. (7) to show the superiority of the former. The focus is on the computation and not on the implementation of the optimum control; the latter is a matter of engineering. The gradient method computes a sequence of perturbations $\delta\phi$ to the control ϕ by

$$\delta\phi \leftarrow -\rho\frac{DJ}{D\phi}(\phi), \tag{28}$$

where ρ is a small enough parameter of descent and D is the *Frechet* derivative [5,22]. The computation of the Frechet derivative is facilitated by the introduction of adjoint variables and the derivation of adjoint equations. This is a very well-known procedure [5,22] and we omit its description here. For all of the three methods, the dynamic programming calculations, the gradient method, and the classical-Newton method, Eq. (22) is transformed to the form of Eq. (10) by means of a Tchebycheff spectral expansion [28] in terms of 64×64 basis functions over the entire spatial domain. The number of basis functions taken is adequate for the resolving, very accurately, all of the pertinent spatial-scales of the hydrodynamics. Time-stepping in dynamic programming is automatically determined by the backward differential formulae. We also mention that in the use of the classical gradient method or the classical-Newton method of determining optimum controls, direct time-integration of Eq. (10) will be required. This is also best achieved by using the BDF [23–26]. The fundamental difference between dynamic programming, and the gradient and classical-Newton methods is that, in the former method, it is the matrix Riccati equation which is integrated in time using the BDF, while in the latter methods it is the state and adjoint equations which are integrated using BDF. This difference is crucial because it does impact significantly on the computational requirements of the last two methods in order to converge to the optimal solution in comparison with dynamic programming.

The optimal control calculations reveal an optimum controlling time-horizon T_f of 2.06 time units. Fig. 1 displays the optimum mobility function obtained from the optimising equation (27) governed by Eqs. (22)–(26). Fig. 2 displays the optimum aqueous fractional flow curve for optimal control. We stress that the aim of this exercise has been to demonstrate the utility of an efficient numerical scheme for computing distributed optimum control functions. The important issue here is that such calculations can be performed to yield insight on the plausibility of control-objectives. The engineering of such controls is entirely another matter and a subject of other research [29].

The optimum control calculations indicate that the hydrodynamic instability is affected by the viscosity ratio of the driving to driven fluid. The fraction of the total driven fluid recovered at breakthrough of the driving fluid increases with decreasing mobility ratio. The optimisation calculations emphasise the sweep efficiency at breakthrough. At infinite mobility ratio, i.e., zero viscosity of the driving fluid, no further recovery of the driven fluid can occur after breakthrough. However, at finite mobility ratio, significant recovery of the driven fluid can continue well after breakthrough. The cumulative recovery of driven fluid as a function of pore volume injected of driving fluid behaves much differently at the finite mobility ratio compared to the infinite mobility ratio. Inhomogeneities at the larger scales will contribute profoundly to diminishing recovery for small mobility ratios. For high but finite mobility

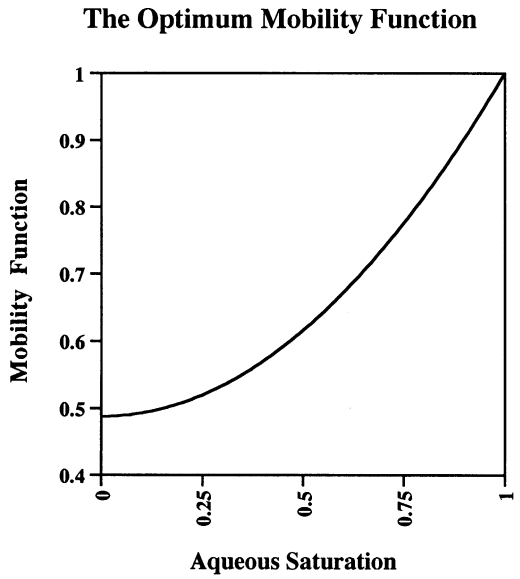

Fig. 1.

ratios the behaviour is different; inhomogeneities on a smaller scale will inhibit recovery significantly more than those on larger scales. The case of the infinite mobility ratio appears to be a singular point, i.e., a point at which the recovery tends to zero as the inhomogeneities become vanishingly small.

In order to assess the efficiency of the three numerical methods, comparisons have been made over 25 different starting guesses for the control function obtained from random sampling over linear feedback control schemes,

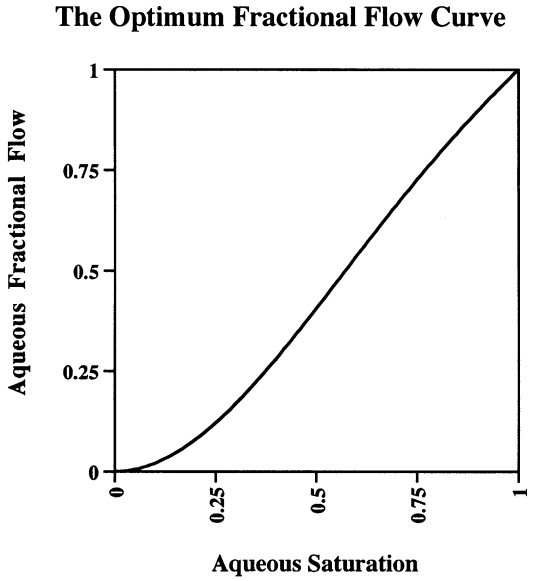

Fig. 2.

Table 1
Comparisons with the gradient-adjoint method; the number of iterations taken to reach optimality

Method	$\Delta J(\phi) \leq 10^{-2}$	$\Delta J(\phi) \leq 10^{-4}$	$\Delta J(\phi) \leq 10^{-6}$
Dynamic programming	33	69	89
Gradient-adjoint method with $\rho = 10^{-2}$	123	771	No convergence
Gradient-adjoint method with $\rho = 10^{-4}$	98	539	No convergence
Gradient-adjoint method with $\rho = 10^{-6}$	53	94	123
Gradient-adjoint method with $\rho = 10^{-8}$	94	617	No convergence

Table 2
Comparisons with the gradient-adjoint method; the CPU times (in minutes) taken to reach optimality on a CRAY/YMP-C90

Method	$\Delta J(\phi) \leq 10^{-2}$	$\Delta J(\phi) \leq 10^{-4}$	$\Delta J(\phi) \leq 10^{-6}$
Dynamic programming	7.1	13.2	18.6
Gradient-adjoint method with $\rho = 10^{-2}$	69.9	494	No convergence
Gradient-adjoint method with $\rho = 10^{-4}$	43.1	280.7	No convergence
Gradient-adjoint method with $\rho = 10^{-6}$	18.6	37.7	62
Gradient-adjoint method with $\rho = 10^{-8}$	43	296	No convergence

i.e. $\lambda(S) \propto S(t, x, y)$; there is no reason to expect that linear feedback will be the optimum control for such a nonlinear problem but these nevertheless form reasonable starting points. Table 1 displays the number of iterations taken to reach optimality by the dynamic programming and gradient-descent methods; the numbers in the table denote the average number of iterations (the decimals have been dropped) required over the 25 starting points for three different convergence criteria pertaining to the first variation $\Delta J(\phi)$ of the objective functional. Four different parameterisations of the gradient method have been considered with respect to the parameter ρ in Eq. (28). The calculations have been carried out on a CRAY-YMP-C90 with 16 processors which is a sequential machine. Table 2 presents the CPU times taken, averaged over the 25 starting points, for the convergence to optimality. Tables 3 and 4 present similar comparisons of the dynamic programming scheme with the classical-Newton method. Note that the number of steps taken by dynamic programming is similar to the number of steps taken by the classical-Newton method. However, the latter requires much more computational time in each step of the iteration.

Finally, we mention some of the technical points related to the computational scheme. Our method for the solution of the matrix Riccati equation using BDF in fact exploits the sparsity of the attendant matrix-kernels. The solution

Table 3
Comparisons with the classical-Newton method; the number of iterations taken to reach optimality

Method	$\Delta J(\phi) \leq 10^{-2}$	$\Delta J(\phi) \leq 10^{-4}$	$\Delta J(\phi) \leq 10^{-6}$
Dynamic programming	33	69	89
Classical-Newton method with optimum step size	40	78	104

Table 4
Comparisons with the classical-Newton method; the CPU times (in minutes) taken to reach optimality on a CRAY/YMP-C90

Method	$\Delta J(\phi) \leq 10^{-2}$	$\Delta J(\phi) \leq 10^{-4}$	$\Delta J(\phi) \leq 10^{-6}$
Dynamic programming	7.1	13.2	18.6
Classical-Newton method with optimum step size	44	76	115.9

Table 5
Comparisons with other Newton methods; the CPU times (in minutes) taken to reach optimality on a CRAY/YMP-C90

Method	$\Delta J(\phi) \leq 10^{-2}$
Dynamic programming	7.1
Quasi-Newton method with the full Broyden–Fletcher-Goldfarb–Shanno update	42
Quasi-Newton method with the limited memory Broyden-Fletcher–Goldfarb–Shanno update	40
Truncated-Newton method based on Lanczos tridiagonalisation	35
Nonlinear conjugate-gradient method	53

of the BDF requires the solution of several intermediate, non-Hermitian, linear systems in the iterative process. These linear systems are treated by us directly using a sparse multifrontal approach [30]. The other approach is to use a pre-conditioned iterative scheme such as the generalised mean residual method (GMRES) based on Krylov sub-spaces [31]. Our investigations have revealed that the direct sparse multifrontal method is far more reliable and more efficient simply because it is not possible to design a general purpose optimum pre-conditioning scheme for the GMRES method applicable to all kinds of hydrodynamic calculations. The presentation of the comparisons of the dynamic programming approach have been restricted only to the gradient-adjoint method and the classical-Newton method. We have in fact investigated other nonlinear optimisation methods such as the quasi-Newton technique based on the Broyden–Fletcher–Goldfarb–Shanno scheme, truncated-Newton, and nonlinear conjugate-gradient methods [32]. These latter methods fare not very well at all in terms of computational time as can be inferred from Table 5.

Our calculations on varying grid sizes indicate that the spectral approximation is exponentially accurate in terms of the accuracy of the functions and functionals being simulated. Tchebycheff expansions are well known to provide exponential convergence insofar as there are no severe (i.e., lowest order) discontinuities in the boundary conditions or the state variables of the hydrodynamic partial differential equations being approximated. Instead of using a spectral approximation by orthogonal collocation via Eqs. (8) and (9) to arrive at the temporal dynamics given by Eq. (10), one can use Karuhnen–Loeve decomposition [2], which is a Galerkin method, to arrive at (10). It is not clear whether Karuhnen–Loeve decomposition possesses exponential convergence over a wide range of initial and boundary conditions on the state variables of the flow-dynamics. Eq. (10) could also be arrived at by the use of *p*- and *hp*-Galerkin finite element discretisation [33]. The advantage of the aforementioned Galerkin finite element discretisations is that they possess exponential convergence like the Tchebycheff spectral methods. The method of determining optimum control functions by the solution of a series of Hamilton–Jacobi equations does not depend on the underlying manner of spatial approximation. Any method of approximation can be used but it is not recommended that methods, such as finite difference spatial approximations, which possess only an algebraic rate of convergence be used as they will necessarily require the solution of much larger dimensional optimal control sub-problems in order to attain the correct optimum control functions.

4. Summary

It is conclusive that there is a significant computational advantage in using the dynamic programming in conjunction with backward differential formulae. It is a method which takes into account the curvature properties of the objective functional. The solution is given in closed-loop form and the closed-loop solution requires the integration in time of a matrix Riccati equation. The information gained in the time-integration is crucial, and although it is the most time consuming step of the overall calculation, it serves to minimise the total number of steps taken in the Newton–Raphson step of Eq. (7). The method is superior to the method of gradients and adjoint equations as well as

the classical method of computing the Newton steps. The method of computing optimum controls presented herein is fully applicable to partial differential equations in any physical or mathematical context. In particular, it holds great promise for performing fully three-dimensional optimum control calculations of fluid flow in heterogeneous porous media [29]. The method can also be applied to the full extension of the Burgers' equation in three dimensions, namely, the equations of Navier–Stokes turbulence. Finally, we note that the Hamilton–Jacobi equation (14) is the infinitesimal form of Bellman's principle of optimality which assures the globally optimal solution *for a given temporal discretisation of the flow-dynamics*. Therefore, Newton steps computed by a sequence of Hamilton–Jacobi equations are the globally optimum corrections allowable at each step of the Newton method for the discretised flow-dynamics. Of course, for the purely continuous flow-dynamics, such a solution need not be strictly global. But, we must note that the determination of optimal control functions on computers, which are of only finite-precision, necessarily requires discretisation of the flow-dynamics, spatially as well as temporally, and there is no way of obviating this aspect. The method developed in this paper has applications in control calculations of fluid dynamics such as in determining the optimal stimulus for the recovery of oil and gas from reservoirs, drag reduction by active control of turbulent flows, the control of yield and specificity of chemical products in chemically reactive turbulent flows by optimum mixing in chemical reactors or even in the turbulent atmosphere where the desired outcome is the minimisation of a reactive pollutant.

Acknowledgements

This work has been supported by the National Science Foundation under grant number NSF/CTS-9504827 and also by the Exxon Research and Engineering Company.

References

[1] B.J. Cantwell, Ann. Rev. Fluid Mech. 13 (1981) 457.
[2] L. Sirovich, Q. Appl. Math. 45 (1987) 561–590.
[3] N. Aubry, P. Holmes, J.L. Lumley, E. Stone, J. Fluid Mech. 192 (1988) 115.
[4] F. Abergel, R. Temam, Theoret. Comput. Fluid Dyn. 1 (1990) 303.
[5] H. Choi, R. Temam, P. Moin, J. Kim, J. Fluid Mech. 253 (1993) 509.
[6] P. Moin, T.R. Bewley, in: Proceedings of the 12th Australasian Fluid Mechanics Conference on Application of Control Theory to Turbulence, Sydney, 1995, pp. 109–117.
[7] C. Lee, J. Kim, H. Choi, J. Fluid Mech. 358 (1998) 245.
[8] C. Min, H. Choi, J. Fluid Mech. 401 (1999) 123.
[9] O. Ghattas, J.-H. Bark, J. Comput. Phys. 136 (1997) 23.
[10] H. Choi, M. Hinze, K. Kunisch, Appl. Numer. Math. 31 (1999) 133.
[11] M. Hinze, Optimal and instantaneous control of the instationary Navier–Stokes equations, Habilitation Thesis, Fachbereich Mathematik, Technische Universität Berlin, 2000.
[12] M. Hinze, K. Kunisch, Three control methods for time-dependent fluid flow, Preprint No. 660/2000, http://www.math.tu-dresden.de/~hinze, Technische Universität Berlin, J. Flow, Contr. Combustion, in press.
[13] G.I. Barenblatt, V.M. Entov, V.M. Ryzhik, Theory of Fluid Flows Through Natural Rocks, Kluwer Academic Publishers, Dordrecht, the Netherlands, 1990.
[14] R. Bellman, Bull. Am. Math. Soc. 60 (1954) 503; see also other Bellman references therein.
[15] R. Aris, Dynamic Programming, Colloquium Lectures in Pure and Applied Sciences, The Socony Mobil Oil Company, Inc., Dallas, TX, 1963.
[16] J.C. Dunn, D.P. Bertsekas, J. Optimization Theory Appl. 63 (1989) 23.
[17] H.H. Rosenbrock, Phys. Lett. A 110 (1985) 343.
[18] H.H. Rosenbrock, Phys. Lett. A 114 (1986) 1.
[19] H.H. Rosenbrock, Phys. Lett. A 114 (1986) 63.
[20] H.H. Rosenbrock, Phys. Lett. A 116 (1986) 410.
[21] H.H. Rosenbrock, Proc. R. Soc. London A 450 (1995) 417.

[22] L.S. Pontryagin, V.G. Boltyanskii, R.V. Gamkrelidze, E.F. Mishenko, The Mathematical Theory of Optimal Processes, Macmillan, New York, 1964.
[23] C.W. Gear, Numerical Initial Value Problems in Ordinary Differential Equations, Prentice-Hall, Englewood Cliffs, NJ, 1971.
[24] P.N. Brown, G.D. Byrne, A.C. Hindmarsh, SIAM J. Sci. Statist. Comput. 10 (1989) 1038.
[25] K.E. Brenan, S.L. Campbell, L.R. Petzold, Numerical Solution of Initial-value Problems in Differential-algebraic Equations, 2nd Edition, SIAM, Philadelphia, PA, 1995.
[26] C. Arévalo, C. Führer, G. Söderlind, ZAMM 77 (8) (1997) 609.
[27] J.A. Sethian, Level Set Methods and Fast Marching Methods, 2nd Edition, Cambridge University Press, Cambridge, 1999.
[28] C. Canuto, M.Y. Hussaini, A. Quarteroni, T.A. Zang, Spectral Methods in Fluid Dynamics, Springer, Berlin, 1988.
[29] P.K. Venkatesh, United States Patent Application in Process (1999).
[30] T.A. Davis, I.S. Duff, SIAM J. Matrix Anal. Appl. 18 (1) (1997) 140.
[31] Y. Saad, M.H. Schultz, SIAM J. Sci. Statist. Comput. 7 (3) (1986) 856.
[32] R. Fletcher, Practical Methods of Optimization, 2nd Edition, Wiley, New York, 2000.
[33] B. Szabo, I. Babuska, Finite Element Analysis, Wiley, New York, 1991.