Supplementary Information for "Finding flows of a Navier-Stokes fluid through quantum computing''

Frank Gaitan

Laboratory for Physical Sciences

College Park, MD

In an effort to make this work as self-contained as possible we briefly describe in this Supplementary Information a number of topics and straight-forward calculations that are needed in the main text. Where appropriate, we point to the literature for further details.

Table of Contents

SI-1. Calorically perfect gas

For many compressible flow applications, the average separation between molecules is sufficiently large that it is safe to ignore intermolecular forces, and pressure and temperature

variations are small enough that specific heats can be treated as constant. These conditions define a calorically perfect gas[1,2]. For such flows the equation of state is the ideal gas law

$$p = \rho R T, \tag{1}$$

where R = 287 J/kg-K is the specific gas constant for air at standard conditions. The specific heat at constant pressure c_p and constant volume c_v satisfy the following well-known relations

$$c_p = \frac{\gamma R}{\gamma - 1} \tag{2a}$$

$$c_v = \frac{R}{\gamma - 1} \tag{2b}$$

$$c_p - c_v = R, \tag{2c}$$

where $\gamma = c_p/c_v$ is the ratio of specific heats. The local internal energy e and enthalpy h are given by

$$\mathrm{e} = c_v T \tag{3a}$$

$$\mathrm{h} = c_p T. \tag{3b}$$

Finally, for an isentropic process, variations in pressure, mass density, and temperature satisfy:

$$\frac{p_2}{p_1} = \left(\frac{\rho_2}{\rho_1}\right)^{\gamma} = \left(\frac{T_2}{T_1}\right)^{\frac{\gamma}{\gamma-1}}. \tag{4}$$

SI-2. Navier-Stokes equations: conservation form

The Navier-Stokes equations for a calorically perfect gas in one spatial dimension were given in the main text as Eqs. (1). As noted there, these equations describe the conservation of mass, momentum, and energy. It is not surprising then that they can be written in conservation form

$$\frac{\partial U}{\partial t} + \frac{\partial F[U,\ \partial U/\partial x]}{\partial x} = J[U], \tag{5}$$

where $U = (U_1, U_2, U_3)^T$, $F = (F_1, F_2, F_3)^T$, and $J = (J_1, J_2, J_3)$ are 3-component column vectors for the flow variables, fluxes, and sources, respectively, and T indicates the transpose operation.

Reading off the arguments of the time derivatives in the Navier-Stokes equations identifies

$$U_1 = \rho; \quad U_2 = \rho v; \quad U_3 = \rho\left(c_V T + \frac{v^2}{2}\right); \tag{6}$$

where we have substituted for the internal energy $\mathrm{e} = c_V T$. Using Eq. (2b), Eq. (6) can be inverted for the flow variables ρ, v, and T in terms of U_1, U_2, and U_3:

$$\rho = U_1; \quad v = U_2/U_1; \quad T = \frac{(\gamma-1)}{2R}\frac{[2U_1U_3 - (U_2)^2]}{(U_1)^2} \ . \tag{7}$$

The pressure p then follows from the equation of state

$$p = \frac{(\gamma-1)}{2}\frac{[2U_1U_3 - (U_2)^2]}{U_1}, \tag{8}$$

and the viscous stress τ is given by

$$\tau = (2\mu + \lambda)\frac{\partial}{\partial x}\left(\frac{U_2}{U_1}\right). \tag{9}$$

Reading off the arguments of the spatial derivatives in the Navier-Stokes equations identifies

$$F_1 = \rho v; \quad F_2 = \rho v^2 + p - \tau; \quad F_3 = \rho v\left(c_V T + \frac{v^2}{2}\right) + pv - \tau v + \kappa\frac{\partial T}{\partial x}.$$

Using Eqs. (6) - (9) in these equations gives

$$F_1 = U_2; \tag{10a}$$

$$F_2 = \left(\frac{3-\gamma}{2}\right)\frac{(U_2)^2}{U_1} + (\gamma-1)U_3 - (2\mu + \lambda)\frac{\partial}{\partial x}\left(\frac{U_2}{U_1}\right); \tag{10b}$$

$$F_3 = \gamma\frac{U_2U_3}{U_1} + \frac{(2-\gamma)}{2}\frac{(U_2)^3}{(U_1)^2} - \frac{(2\mu+\lambda)}{2}\frac{\partial}{\partial x}\left[\left(\frac{U_2}{U_1}\right)^2\right] + \frac{\kappa(\gamma-1)}{2R}\frac{\partial}{\partial x}\left[\frac{2U_1U_3 - (U_2)^2}{(U_1)^2}\right]. \tag{10c}$$

As this accounts for all terms in the Navier-Stokes equations we see that the source term vanishes:

$$J = 0. \tag{11}$$

Eqs. (6), (10), and (11) thus specify U, F[U,∂U/∂x], and J[U] and we arrive at the desired conservation form for the Navier-Stokes equations in one spatial dimension.

SI-3 Navier-Stokes equations: finite difference approximation

As with all computational fluid dynamics calculations, it is necessary to discretize the Navier-Stokes dynamics[3,4,5]. We do this by replacing the spatial continuum parameterized by x with a spatial grid of m points $\{x_1, \ldots, x_j, \ldots, x_m\}$. The grid-points are chosen so that $\Delta x = x_{j+1} - x_j$ is independent of j. Prior to discretization the flow variables {…, U(x,t), …} constitute an uncountable number of degrees of freedom labeled by x. Discretization reduces this to a finite number of degrees of freedom {U(1,t), …, U(j,t), …, U(m,t)}, where $U(j,t) \equiv U(x_j, t)$. We replace first and second spatial derivatives with a central finite difference approximation,

$$\left.\frac{\partial U}{\partial x}\right|_{x_j} = \frac{U(j+1,t) - U(j-1,t)}{2\Delta x} + \mathcal{O}(\Delta x^2) \qquad (12a)$$

$$\left.\frac{\partial^2 U}{\partial x^2}\right|_{x_j} = \frac{U(j+1,t) - 2U(j,t) + U(j-1,t)}{(\Delta x)^2} + \mathcal{O}(\Delta x^2), \qquad (12b)$$

which is a second order approximation in Δx. The Navier-Stokes equations in one spatial dimension (see Eq. (5)) require the spatial derivative of the flow flux F = F[U,∂U/∂x]. Differentiating F gives

$$\left.\frac{\partial F}{\partial x}\right|_{x_j} = \mathcal{F}[\partial U/\partial x\,, \partial^2 U/\partial x^2]\,, \qquad (13)$$

where the exact form of $\mathcal{F}$ follows from differentiating Eqs. (10). Replacing ∂U/∂x and ∂^2U/∂x^2 in Eq.(13) by Eqs. (12) gives

$$\left.\frac{\partial F}{\partial x}\right|_{x_j} = -f(U(j+1,t), U(j,t), U(j-1,t)\,) \equiv -f(U), \qquad (14)$$

where the exact form of f(U) follows from implementing this substitution. Note that if a source term J[U] had been present it would be unchanged by discretization and so it could be absorbed into f(U). Recall from Section SI-2 that only $F[U, \partial U/\partial x]$ contains terms that depend on the viscosity coefficients, with these terms being present (absent) for viscous (inviscid) flow. Thus, the presence or absence of viscosity only affects the specific algebraic expression for $F[U, \partial U/\partial x]$, and the same is true for f(U). We will return to this point below. We see that discretizing the Navier-Stokes partial differential equation (PDE) via the above finite difference approximation reduces it to a nonlinear set of ordinary differential equations (ODE)

$$\frac{dU}{dt} = f(U). \qquad (15)$$

Note that our procedure for discretizing space is just one possible choice. Alternative discretization procedures are possible and each would lead to a different expression for the driver function f(U). The alternative driver function for a given discretization would be substituted for our driver function in the quantum algorithm described in Section SI-4. This raises the interesting question of how different spatial discretization procedures impact the stability of that quantum algorithm. We stress that although we have focused on the Navier-Stokes PDE in one spatial dimension, discretizing an arbitrary set of PDEs in a finite number of spatial dimensions using an appropriate generalization of the above finite difference approximation would also reduce it to a set of ODEs.

SI-4. Solving Navier-Stokes ODE using quantum ODE solver

We summarize the application of Kacewicz' quantum ODE algorithm[6] to the Navier-Stokes ODE. The goal is to compute a bounded function *l(j,t)* which approximates the exact solution

U(j,t) of Eq. (15) over a time-interval $0 \leq t \leq T$ subject to the initial condition $l(j,0) = U(j,0) = U_0(j)$. Here f(U) and U are 3-component column vectors denoting the ODE driver function and flow variables, respectively. As in the main text, j labels the spatial grid-points $\{x_j \mid 1 \leq j \leq m\}$ introduced in the discretization of the Navier-Stokes dynamics.

Ref. 6 assumes the driver function f(U) has r continuous, bounded partial derivatives, with the components of the r^{th} derivatives satisfying the Hölder condition

$$|\partial^r f^s(U) - \partial^r f^s(U')| \leq H\,||U - U'||^\rho \ . \qquad (16)$$

Here: $1 \leq s \leq 3$ labels the components of f(U); $\partial^r = \partial_1^a \partial_2^b \partial_3^c$ with a+b+c = r; H > 0; ||v|| is the maximum norm which is the largest of $\{|v_1|, |v_2|, |v_3|\}$; and $0 < \rho \leq 1$. The smoothness of the driver f(U) is parameterized by $q = r + \rho > 0$. Functions satisfying these conditions are known as Hölder class functions[7,8] and are elements of the Hölder space $F^{r,\rho}$.

The algorithm begins by partitioning the time interval $0 \leq t \leq T$ into n subintervals $I_i = [t_i, t_{i+1}]$ of duration h = T/n using n+1 intermediate times $\{t_0 = 0, \ldots, t_n = T\}$. Each subinterval I_i is further subdivided into $N_k = n^{k-1}$ sub-subintervals of duration $\bar{h} = h/N_k = T/n^k$ using $N_k + 1$ times $\{t_i^0 = t_i, \ldots, t_i^{N_k} = t_{i+1}\}$. The l^{th} sub-subinterval is then $I_i^l = [t_i^l, t_i^{l+1}]$. At each primary partition time t_i and grid-point j, parameters $y_i(j)$ are introduced which: (i) approximate the exact solution at these times $U(j,t_i) \approx y_i(j)$; and (ii) initialize the approximate solution in each subinterval (see below). How n, k, and $y_i(j)$ are assigned values is described below. Note that in Refs. 18 and 19, Kacewicz introduced, respectively, 0 and n time sub-subintervals and found sub-optimal algorithms. As seen in the computational complexity discussion in the paper accompanying this SI, the above partitioning into n^{k-1} sub-subintervals per subinterval gives rise to an (almost) optimal quantum algorithm.

In each sub-subinterval I_i^l, Taylor's method[9,10,11] is used to approximate the exact solution U(j,t) by a truncated Taylor series about t_i^l:

$$l_{i,l}(j,t) = l_{i,l}(j,t_i^l) + \sum_{\nu=0}^{r} \frac{1}{\rho!} \frac{d^\nu f(j,t_i^l)}{dt^\nu} (t - t_i^l)^\nu + \mathcal{O}(\bar{h}^{r+1}). \qquad (17)$$

For Hölder class functions in $F^{r,\rho}$, r is given. For a quasi-smooth driver function f(U) (as will be considered in the main text) r is chosen so that the $\mathcal{O}(\bar{h}^{r+1})$ error is sufficiently small. The approximate solution $l_i(j,t)$ for subinterval I_i is defined by requiring that: (i) the $y_i(j)$ provide the initial condition $l_{i,0}(j,t_i^0) = y_i(j)$; and (ii) the $\{l_{i,l}(j,t) \mid 0 \leq l \leq N_k - 1\}$ join continuously at the common boundary time of adjacent sub-subintervals: $l_{i,l+1}(j,t_i^{l+1}) = l_{i,l}(j,t_i^{l+1})$. Kacewicz then defines $l_i(j,t) = l_{i,l}(j,t)$ when $t \in I_i^l$ and $0 \leq l \leq N_k - 1$. Thus, once $y_i(j)$ is known, $l_i(j,t)$ is determined throughout I_i.

To determine the parameters $\{y_i(j)\}$ Kacewicz integrates Eq. (15) over I_i giving

$$U(j,t_{i+1}) = U(j,t_i) + \sum_{l=0}^{N_k-1} \int_{t_i^l}^{t_i^{l+1}} d\tau\ f(l_{i,l}(\tau)) + \sum_{l=0}^{N_k-1} \int_{t_i^l}^{t_i^{l+1}} d\tau \big[f(U(j,\tau) - f(l_{i,l}(j,\tau)\big],$$

which is exact. To obtain a relation among the $\{y_i(j)\}$ he discards the third term on the RHS; replaces $U(j,t_i)$, $U(j,t_{i+1})$ with $y_i(j)$, $y_{i+1}(j)$, respectively; and writes $\tau = \bar{h}u$ giving

$$y_{i+1}(j) = y_i(j) + N_k \sum_{l=0}^{N_k-1} \frac{\bar{h}}{N_k} \int_0^1 du\, f\left(l_{i,l}(j,u)\right) \tag{18}$$

for $0 \leq i \leq n-1$. Eq. (18) determines $y_{i+1}(j)$ from $y_i(j)$ and the Taylor polynomials $\{l_{i,l}(j,u)\}$. Determining the $\{y_i(j)\}$ begins by setting $y_0(j)$ to the ODE initial condition: $y_0(j) = U_0(j)$. The $\{y_0(j)\}$ then determine the approximate solution $l_0(j,t)$ throughout $I_0 = [0, t_1]$ as just discussed. Proceeding to $I_1 = [t_1, t_2]$, Eq. (18) determines $y_1(j)$ from $y_0(j)$ once the integral on the RHS is evaluated. To that end Kacewicz introduces N_k knot times $\{u_v^l\}$ in each sub-subinterval I_i^l and approximates the integral with the average of its integrand over the knot times:

$$\mathcal{I}_i(j) \stackrel{\text{def}}{=} \frac{\bar{h}}{N_k} \sum_{l=0}^{N_k-1} \int_0^1 du\, f(l_{i,l}(j,u)) \approx \frac{\bar{h}}{{N_k}^2} \sum_{l,v=0}^{N_k-1} f\left(l_{i,l}(j,{u_v}^l)\right). \tag{19}$$

Estimating the average value of f is the only task in Kacewicz' quantum ODE algorithm that requires a quantum computer. As explained in Section SI-3, for the Navier-Stokes equations, whether a flow is viscous or not only impacts the algebraic expression for f(U). For either type of flow, the QAEA takes the relevant form of f(U) and returns an estimate of the average value of f in Eq. (19). The quantum algorithm thus works for both viscous and inviscid flows. Note that, before the quantum amplitude estimation algorithm[12] (QAEA) can be used to estimate the average of f, we must shift and rescale it so that each of its component f^s takes values in the range [0,1]. Let $f_{max}{}^s$ ($f_{min}{}^s$) = max (min)$\{f^s(l_{i,l}(j,u_v^l)| \, 0 \leq l,v \leq N_k - 1\}$ and $\Delta f^s = f_{max}{}^s - f_{min}{}^s$. Then defining

$$g^s\left(l_{i,l}(j,{u_v}^l)\right) = \frac{f^s\left(l_{i,l}(j,{u_v}^l)\right) - {f_{min}}^s}{\Delta f^s}$$

insures $g^s(l_{i,l}(j,u_v^l))$ takes values in the range [0,1]. With these definitions Eq. (19) becomes (suppressing the superscript *s*)

$$\mathcal{I}_i(j) \approx \bar{h} f^{min} + \bar{h} \Delta f \mathcal{G}_i(j), \tag{20}$$

with

$$\mathcal{G}_i(j) = \frac{1}{{N_k}^2} \sum_{l,v=0}^{N_k-1} g\left(l_{i,l}(j,{u_v}^l)\right). \tag{21}$$

The QAEA returns an approximate value $\tilde{\mathcal{G}}_i(j)$ for $\mathcal{G}_i(j)$ that satisfies the bound

$$\left|\tilde{\mathcal{G}}_i(j) - \mathcal{G}_i(j)\right| < \varepsilon_1 \tag{22}$$

with probability $1 - \delta_1$ (when the algorithm's success probability is boosted by repeated application[13]). Eq. (20) then gives an estimate $\tilde{\mathcal{I}}_i(j)$ for $\mathcal{I}_i(j)$ which satisfies Eq. (22) with

$\varepsilon_1 \rightarrow (\bar{h}\Delta f)\varepsilon_1$. Starting with I_0, the $\{y_1(j)\}$ are found from $\{y_0(j)\}$ and the QAEA. The $\{y_1(j)\}$ then determine the approximate solution $l_1(j,t)$ in $I_1 = [t_1, t_2]$. Iterating this procedure over the remaining subintervals I_i gives the algorithm's approximate solution $l(j,t)$, where

$$l(j,t) = l_i(j,t)$$

for $t \in I_i$ and $0 \leq i \leq n-1$. Ref. 6 shows that for Hölder class driver functions $f(U) \in \mathcal{F}^{r,\rho}$ the error ϵ in $l(j,t)$ satisfies (for $0 \leq t \leq T$, $1 \leq j \leq m$, and $n \geq 5$) the upper bound

$$\epsilon \equiv sup\|U(j,t) - l(j,t)\| = \mathcal{O}\left(\frac{1}{n^{\alpha_k}}\right),$$

with probability $1 - \delta$. Here $\alpha_k = k(q+1) - 1$ and $q = r + \rho$. To achieve this performance Kacewicz requires $\varepsilon_1 = 1/n^{k-1}$ and $1 - \delta_1 = (1 - \delta)^{1/n^k}$. Later, when discussing our test application for the quantum Navier-Stokes algorithm, we explain how n and k are chosen for given ε_1 and δ which completes the specification of our quantum PDE algorithm.

SI-5. Quantum simulation of steady, inviscid, compressible flow through a de Laval nozzle

As a test application for our quantum Navier-Stokes algorithm we considered the steady, inviscid, compressible flow of air through a convergent-divergent (de Laval) nozzle (see main text Figure 1). Such nozzles are used in rocket engines and in jet engines for supersonic flight. For flow temperatures below 1000 K air can be treated as a calorically perfect gas[2] and so we make that assumption. The nozzle shape is characterized by its cross-sectional area A(x) which decreases (convergent section) as we move from the nozzle entrance to the throat, and then increases (divergent section) as we move on to the nozzle exit. The flow can be supersonic in the divergent section and a normal shockwave can appear there for a wide range of (nozzle) exit pressures. Except inside a shockwave (and the small boundary layer near the nozzle surface), dissipative effects are negligible and the inviscid flow is isentropic.

In Section SI-5A we define the quasi-1D approximation for nozzle flow which applies when A(x) varies sufficiently slowly. We then describe the derivation of the Navier-Stokes equations that are consistent with this approximation and write them in a form suitable for numerical simulation. In Section SI-5B we describe how the *exact* flow solution is found when a shockwave is and is not present in the nozzle. In Sections SI-5C and -5D we present, respectively, the boundary and initial conditions that are appropriate for this flow problem. Finally, in Section SI-5E we explain how the time partition parameters n and k are chosen. Note that many topics presented in this Section are standard[1,2,5] — they are included solely to make the paper more self-contained.

SI-5A Quasi-1D approximation

Inviscid flow through a de Laval nozzle must be tangent to its surface and so the exact flow solution is two-dimensional. However, if the nozzle area A(x) varies sufficiently slowly, it is

often a good approximation to assume the flow variables depend solely on position x along the nozzle and time t. The flow is primarily in the x-direction except near the nozzle surface where it is tangential. This is the *quasi-1D approximation*.

The Navier-Stokes PDE presented in the main text is for strictly 1D flows. This is not exactly true for quasi-1D flows and so it is necessary to derive Navier-Stokes equations that are consistent with this approximation. This is done in Ref. 5 which we follow here. Integrating the exact 2D Navier-Stokes equations over a control volume of thickness dx, and invoking the assumptions of the quasi-1D approximation gives the following set of PDEs:

$$\frac{\partial}{\partial t}(\rho A) + \frac{\partial}{\partial x}(\rho A v) = 0 \qquad (23a)$$

$$\frac{\partial}{\partial t}(\rho A v) + \frac{\partial}{\partial x}\left(\rho A v^2 + \frac{1}{\gamma} p A\right) = \frac{1}{\gamma} p \frac{\partial A}{\partial x} \qquad (23b)$$

$$\frac{\partial}{\partial t}\left[\rho A\left(\frac{e}{\gamma - 1} + \frac{\gamma}{2} v^2\right)\right] + \frac{\partial}{\partial x}\left[\rho A v\left(\frac{e}{\gamma - 1} + \frac{\gamma}{2} v^2\right) + p A v\right] = 0. \qquad (23c)$$

These are the Navier-Stokes equations in the quasi-1D approximation. For numerical simulation purposes it proves convenient to switch over to *dimensionless* variables. Let p_0, ρ_0, and T_0 denote the values of pressure, mass density, and temperature at the nozzle entrance. Define the *dimensionless* pressure $p' = p/p_0$, mass density $\rho' = \rho/\rho_0$, and temperature $T' = T/T_0$. Let L denote the nozzle length and $a_0 = \sqrt{(\gamma R T_0)}$ the speed of sound at the nozzle entrance. Define dimensionless position $x' = x/L$, time $t' = a_0 t/L$, and velocity $v' = v/a_0$. The local Mach number is then $M = v/a = v'/a'$, where $a' = a/a_0$ is the dimensionless local speed of sound. Finally, define the dimensionless nozzle area $A' = A/A_t$, where A_t is the area at the nozzle throat. Since Eqs. (23) have well-defined dimensions, switching to dimensionless variables does not alter their form. When discussing the numerical simulation of Eqs. (23), we *always* assume they are expressed in *dimensionless* variables and will *suppress* the primes that indicate such variables.

Eqs. (23) take the conservation form of the main text's Eqs. (2) with:

$$U_1 = \rho A; \qquad U_2 = \rho A v; \qquad U_3 = \rho A\left(\frac{e}{\gamma - 1} + \frac{\gamma}{2} v^2\right); \quad (24a)$$

$$F_1 = U_2; \qquad F_2 = \frac{{U_2}^2}{U_1} + \frac{\gamma - 1}{\gamma}\left(U_3 - \frac{\gamma}{2}\frac{{U_2}^2}{U_1}\right); \qquad F_3 = \frac{\gamma U_2 U_3}{U_1} - \frac{\gamma(\gamma - 1)}{2}\frac{{U_2}^3}{{U_1}^2}; \quad (24b)$$

$$J_1 = 0; \qquad J_2 = \frac{\gamma - 1}{\gamma}\left(U_3 - \frac{\gamma}{2}\frac{{U_2}^2}{U_1}\right)\frac{\partial}{\partial x}(\ln A); \qquad J_3 = 0. \qquad (24c)$$

Given the solution variables $U = (U_1, U_2, U_3)$, the *dimensionless* primary flow variables are:

$$\rho = \frac{U_1}{A}; \quad v = \frac{U_2}{U_1}; \quad T = (\gamma - 1)\left(\frac{U_3}{U_1} - \frac{\gamma}{2}\frac{{U_2}^2}{{U_1}^2}\right); \quad p = \rho T; \quad e = T. \qquad (25)$$

Finally, we determine the ODE driver function f(U). We begin with the conservation form of the Navier-Stokes PDE

$$\frac{\partial U}{\partial t} = -\frac{\partial}{\partial x}F(U) + J(U),$$

where F does not depend on ∂U/∂x as we are considering inviscid flow. Making a central finite difference approximation for ∂F/∂x gives the ODE

$$\frac{d}{dt}U(j,t) = -\frac{1}{2\Delta x}\left[F\big(U(j+1,t)\big) - F\big(U(j-1,t)\big)\right] + J\big(U(j,t)\big) \equiv f\big(U(j,t)\big), \quad (26)$$

which identifies

$$f_1\big(U(j,t)\big) = -\frac{1}{2\Delta x}\left[F_1\big(U(j+1,t)\big) - F_1\big(U(j-1,t)\big)\right] \quad (27a)$$

$$f_2\big(U(j,t)\big) = -\frac{1}{2\Delta x}\left[F_2\big(U(j+1,t)\big) - F_2\big(U(j-1,t)\big)\right] + J_2\big(U(j,t)\big) \quad (27b)$$

$$f_3\big(U(j,t)\big) = -\frac{1}{2\Delta x}\left[F_3\big(U(j+1,t)\big) - F_3\big(U(j-1,t)\big)\right]. \quad (27c)$$

SI-5B Exact solution

The exact solution for steady, inviscid, compressible flow through a de Laval nozzle is derived in Refs. 2 and 5. Our presentation follows theirs. Note that we use *dimensionful* variables in this subsection. For isentropic flow we have:

$$\left(\frac{A}{A_*}\right)^2 = \frac{1}{M^2}\left[\frac{2}{\gamma+1}\left(1 + \frac{\gamma-1}{2}M^2\right)\right]^{(\gamma+1)/(\gamma-1)} \quad (28a)$$

$$\frac{p}{p_0} = \left(1 + \frac{\gamma-1}{2}M^2\right)^{-\gamma/(\gamma-1)} \quad (28b)$$

$$\frac{\rho}{\rho_0} = \left(1 + \frac{\gamma-1}{2}M^2\right)^{-1/(\gamma-1)} \quad (28c)$$

$$\frac{T}{T_0} = \left(1 + \frac{\gamma-1}{2}M^2\right)^{-1}. \quad (28d)$$

Here the 0-subscript on a flow variable indicates its stagnation value which is the value it would have if the associated fluid element were brought to rest isentropically from its actual state of motion. For our flow problem, in the *absence* of a shockwave, the stagnation value is effectively the value at the nozzle entrance. Here A_* is the area of the nozzle where the flow is sonic if its associated fluid element at x were isentropically accelerated to M = 1. In the *absence* of a

shockwave the flow throughout the nozzle is isentropic and so A_* is the area of the throat A_t and Eqs. (28) give the exact flow solution.

In the presence of a shockwave in the divergent part of the nozzle, Eqs. (28) give the flow *ahead* of the shockwave. *Behind* the shockwave things are more complicated as the stagnation values of the flow variables are no longer the nozzle entrance values. Let p_{0_1} (p_{0_2}) denote the stagnation pressure before (behind) the shockwave. The exact isentropic flow solution *behind* the shockwave is:

$$\left(\frac{A}{A_*}\right)^2 = \frac{1}{M^2}\left[\frac{2}{\gamma+1}\left(1+\frac{\gamma-1}{2}M^2\right)\right]^{(\gamma+1)/(\gamma-1)} \tag{29a}$$

$$\frac{p}{p_0} = \frac{p_{0_2}}{p_{0_1}}\left(1+\frac{\gamma-1}{2}M^2\right)^{-\gamma/(\gamma-1)} \tag{29b}$$

$$\frac{\rho}{\rho_0} = \frac{p_{0_2}}{p_{0_1}}\left(1+\frac{\gamma-1}{2}M^2\right)^{-1/(\gamma-1)} \tag{29c}$$

$$\frac{T}{T_0} = \left(1+\frac{\gamma-1}{2}M^2\right)^{-1}, \tag{29d}$$

where p_0, ρ_0, and T_0 are the pressure, density, and temperature at the *nozzle entrance*. Note that for positions *behind* the shockwave A_* is no longer the throat area. To determine the exact solution behind the shockwave it is necessary to determine the shockwave location x_1 in the divergent section of the nozzle. Once this is done, Eqs. (28) and (29) determine the exact flow solution as we shall see.

For a given nozzle profile A(x) the location of the shockwave is controlled by the exit pressure p_e. Ref. 2 derives the following expression for the exit Mach number M_e:

$${M_e}^2 = -\frac{1}{\gamma-1} + \sqrt{\left(\frac{1}{(\gamma-1)^2} + \frac{2}{\gamma-1}\left(\frac{2}{\gamma+1}\right)^{(\gamma+1)/(\gamma-1)}\left(\frac{p_{0_e}}{p_e}\frac{{A_e}^*}{A_e}\right)^2\right)}, \tag{30}$$

where, at the nozzle exit, p_{0_e} is the stagnation pressure, A_e is the nozzle area, and ${A_e}^*$ is the sonic nozzle area. Ref. 2 shows that

$$\frac{p_e A_e}{p_{0_e}{A_e}^*} = \frac{p_e}{p_0}\frac{A_e}{A_t}, \tag{31}$$

where p_0 is the nozzle entrance pressure and A_t is the throat area. In our numerical simulations p_e/p_0 and A_e/A_t are known and $\gamma = 1.4$. Eqs. (30) and (31) then determine the exit Mach number M_e. Eq. (28b) determines p_e/p_{0_e} since p_{0_e} is the stagnation pressure for the exit flow. The ratio of stagnation pressures p_{0_2}/p_{0_1} satisfies[2]

$$\frac{p_{0_2}}{p_{0_1}} = \frac{p_{0_e}}{p_e}\frac{p_e}{p_0}. \tag{32}$$

Since p_e/p_{0_e} and p_e/p_0 are both known, Eq. (32) determines p_{0_2}/p_{0_1}. Ref. 5 derives the following relation between this ratio and the Mach number M_1 just *before* the shockwave:

$$\frac{p_{0_2}}{p_{0_1}} = \left[\frac{(\gamma+1){M_1}^2}{(\gamma-1){M_1}^2+2}\right]^{\gamma/(\gamma-1)}\left[\frac{\gamma+1}{2\gamma{M_1}^2-(\gamma-1)}\right]^{1/(\gamma-1)}. \qquad (33)$$

Solving for M_1^2 gives a polynomial equation and M_1^2 is the real, positive root that is greater than 1 (flow is supersonic just before the shockwave). Given M_1^2, Eq. (28a) gives the nozzle area A_1/A_t at the shockwave location x_1, and knowing the nozzle area profile A(x) gives x_1.

As already noted, for x ahead of the shockwave, Eqs. (28) determine the local flow variables at x. For x in the convergent section of the nozzle, the flow is subsonic. Plugging A(x) into Eq. (28a) and solving for the real, positive root that is less than 1 gives $M^2(x)$. Eqs. (28b)-(28d) then give the remaining primary flow variables at x. For x in the divergent section of the nozzle, but ahead of the shockwave, we again plug A(x) into Eq. (28a), only now we solve for the real, positive root that is greater than 1 since the flow is supersonic here. This gives $M^2(x)$ which is plugged into Eqs. (28b)-(28d) to determine the remaining primary flow variables. For x *behind* the shockwave the flow is subsonic. Plugging A(x) into Eq. (29a) and solving for the real, positive root less than 1 gives $M^2(x)$, and Eqs. (29b)-(29d) give the remaining primary flow variables.

This determines the exact flow solution when a shockwave is absent and present. These results are compared with the output of our numerical simulation of the quantum Navier-Stokes algorithm applied to this nozzle flow problem.

SI-5C Boundary conditions

In this subsection all flow variables are *dimensionless*. For unsteady, inviscid flows the Navier-Stokes equations are hyperbolic PDEs[5]. For such PDEs the method of characteristics is used to establish the appropriate boundary conditions for their solution[3,5]. We summarize the conclusions reached in Ref. 5 for nozzle flow problems:

(i) *subsonic inflow boundary* — two flow variables must be assigned values and one allowed to float (see below);
(ii) *subsonic outflow boundary* — one flow variable must be assigned a value and two allowed to float;
(iii) *supersonic outflow boundary* — three flow variables are allowed to float.

The boundary value of a floating variable is specified using linear extrapoltation[5] which uses the values of this variable at its two nearest interior grid-points to assign the boundary value.

In our numerical simulation the nozzle entrance is always a subsonic inflow boundary. The entrance grid-point label is j = 1 and that of its two nearest interior grid-points are j = 2, 3. At the entrance ρ(1,t) = T(1,t) = 1 by our definition of dimensionless variables and we use these two variables as the fixed-valued variables. The mass flow rate $U_2 = \rho A v$ will vary as the simulation

proceeds to the final steady-state and so it is chosen as the floating variable. The boundary conditions at the nozzle entrance are[5]:

$$U_1(1,t) = A(1);$$

$$U_2(1,t) = 2U_2(2,t) - U_2(3,t);$$

$$U_3(1,t) = U_2(1,t)\left[\frac{1}{\gamma-1} + \frac{\gamma}{2}\left(\frac{U_2(1,t)}{U_1(1,t)}\right)^2\right].$$

In the absence of a shockwave the nozzle exit is a supersonic outflow boundary. Thus all components of U must be floating variables. The nozzle exit grid-point label is j = m and its two nearest interior grid-points have labels j = m-1, m-2. The boundary conditions at the exit are[5]:

$$U_a(m,t) = 2U_a(m-1,t) - U_a(m-2,t) \qquad (a = 1,2,3).$$

Finally, when a shockwave is present, the nozzle entrance remains a subsonic inflow boundary and is handled as above. The nozzle exit is now a subsonic outflow boundary. Since the exit pressure p_e controls the shockwave location and is not varied, the simulation uses it as the single fixed-valued variable. We use U_1 and U_2 as the two floating variables. The boundary condition at the nozzle exit is[5]:

$$U_1(m,t) = 2U_1(m-1,t) - U_1(m-2,t)$$

$$U_2(m,t) = 2U_2(m-1,t) - U_2(m-2,t)$$

$$U_3(m,t) = \frac{p_e A_e}{\gamma-1} + \frac{\gamma}{2}\frac{(U_2(m,t))^2}{U_1(m,t)}.$$

In all cases $\gamma = 1.4$.

SI-5D Initial condition

All variables in this subsection are *dimensionless* and primes superscripts are suppressed. The time propagation procedure used in our quantum PDE algorithm is an explicit procedure which uses a continuity requirement to propagate the Taylor polynomials across a subinterval I_i, and the initial value y_{i+1} for subinterval I_{i+1} is determined from y_i and the Taylor polynomials for I_i. Such explicit procedures are known to have conditional numerical stability[3,5]. If the initial condition for the numerical simulation deviates too much from the exact solution, the simulation becomes unstable and eventually crashes. Although this can be ameliorated somewhat by increasing the number of grid-points, the simulation runtime increases and so there are practical limits to how far the initial condition can deviate from the exact solution. To account for this in a systematic way we chose our initial condition to be a random shift of the exact steady-state flow solution found in SI-5B. The details are as follows.

We begin with the steady-state flow solution for the mass density $\rho_{ss}(j)$, temperature $T_{ss}(j)$, and mass flow rate $\dot{m}_{ss} = \rho_{ss} A v_{ss}$. Because of mass conservation the mass flow rate is constant

across the nozzle. It can be easily evaluated at the throat using Eqs. (28c) and (28d), and that the throat velocity is the local speed of sound $v_t = \sqrt{T}$ and the nozzle throat area $A_t = 1$. The result is $\dot{m}_{ss} = 0.579$. Note that the flow variable $U_2 = \rho A v$ is the mass flow rate. To the mass density and temperature we add a random shift at each grid-point j, and to the mass flow rate a global random shift:

$$\rho_{init}(j) = \rho_{ss}(j) + \delta\rho(j)$$
$$T_{init}(j) = T_{ss}(j) + \delta T(j)$$
$$\dot{m}_{init} = 0.579 + \delta\dot{m}.$$

The random shifts for mass density and temperature are drawn independently at each grid-point j from a uniform distribution. For example, for the temperature, $\delta T(j)$ is drawn uniformly from the interval $[-\delta T_{max}, \delta T_{max}]$, where $\delta T_{max} = 0.02 T_{ss}^{min}$ ($0.01 T_{ss}^{min}$) and T_{ss}^{min} is the smallest value $T_{ss}(j)$ takes over all j when a shockwave is absent (present). Similarly, for the mass density $\delta\rho(j)$ is 10% (2%) of the smallest value of $\rho_{ss}(j)$. For the mass flow rate the shift $\delta\dot{m}$ is 1% of its steady-state value. The initial condition for the simulation takes $\rho_{init}(j)$, $T_{init}(j)$, and $\dot{m}_{init}$ and plugs them into the formulas for U, F, J, and f(U) appearing in SI-5A. The latter values are then used to initiate the numerical simulation.

SI-5E Choosing time partition parameters n and k

All variables in this subsection are *dimensionless*. For linear hyperbolic PDEs, a numerical stability analysis gives rise to a relation between the local time-step Δt_j at grid-point j in a time-marching simulation algorithm and the grid spacing Δx:

$$\Delta t_j = \frac{C\Delta x}{a_j + v_j} \qquad (j = 1, \ldots, m)\ . \qquad (34)$$

Here C is the Courant number which we set equal to ½; a_j is the local speed of sound, and v_j is the local flow velocity. This is the Courant-Friedrichs-Lewy (CFL) stability criterion[14]. Strictly speaking it does not apply to nonlinear PDEs such as the Navier-Stokes equations, though it usually provides good guidance. A global time-step Δt_{CFL} is defined as the minimum value taken by the $\{\Delta t_j\}$. In our numerical simulation we determine Δt_{CFL} using the simulation's initial condition which allows Eq. (34) to be evaluated for all j. The simulation runs over the time interval [0, T] and we define $T = N_{tot}\,\Delta t_{CFL}$, where N_{tot} is an input to the simulation. In our partitioning of [0, T] we introduced n^k sub-subintervals I_i^l of duration $\bar{h}$ so that $T = \bar{h} n^k$. From these two expressions for T we find

$$\frac{\bar{h}}{\Delta t_{CFL}} = \frac{N_{tot}}{n^k}.$$

We require $\bar{h}/\Delta t_{CFL} < 1$ so that our time partition is compatible with the CFL stability criterion. We choose n and k so that this is true.

To that end, recall that we required the error ϵ_1 in the estimate of the average of f(U) to satisfy $\epsilon_1 = 1/n^{k-1}$. Solving for k gives

$$k = 1 + \lceil \ln(1/\epsilon_1)/\ln(n) \rceil . \quad (35)$$

We choose n and k using the following iterative procedure. The simulation accepts as inputs N_{tot}, ϵ_1, and a guess n_{in} for n. Set $n_0 = n_{in}$ and k_0 to satisfy Eq. (35) using n_0. Evaluate the ratio $N_{tot}/n_0^{k_0}$. If the ratio is less than 1, then $\bar{h}/\Delta t_{CFL} < 1$ and we accept n_0 and k_0 as n and k. Otherwise, increment $n_i = n_{i-1} + 1$ and evaluate k_i from Eq. (35). Re-evaluate the ratio $N_{tot}/n_i^{k_i}$ and repeat this procedure until the current ratio $N_{tot}/n_a^{k_a}$ is first less than 1. Accept n_a and k_a as n and k and use them to partition the time interval [0, T].

SI-6. Quantum amplitude estimation

In this Section we summarize (for the reader's convenience) the arguments of Brassard et al.[12] which lead to the Quantum Amplitude Estimation Algorithm (QAEA); and following Novak[15], show how this algorithm can be used to estimate the mean-value of a function.

The outline of our discussion is as follows: (i) Section SI-6A presents results from quantum amplitude amplification[12] that are relevant to the QAEA; (ii) Section SI-6B then explains how these results are combined with quantum phase estimation[16] to yield the QAEA[12]; and (iii) Section SI-6C explains how the QAEA can be used to estimate the function mean[15].

SI-6A Quantum amplitude amplification

The quantum amplitude amplification algorithm[12] generalizes Grover's quantum search algorithm[17]. The Hilbert space $\mathcal{H}$ is thus assumed to partition into two subspaces $\mathcal{H}_1$ and $\mathcal{H}_0$ known, respectively, as the ``good'' and ``bad'' subspaces. Without loss of generality, the computational basis (CB) {|j>: $0 \leq j < M$} is also assumed to partition into two subsets, one spanning $\mathcal{H}_1$ and the other $\mathcal{H}_0$. To specify this partition, a Boolean function χ: {0, …, M-1} ⟶ {0,1} is introduced so that |j> is a good (bad) basis state if $\chi(j) = 1$ (0).

Suppose we have a quantum algorithm $\mathcal{A}$ that implements a unitary operator A. If the CB state |0> is used as the initial state for $\mathcal{A}$, then the final state |ψ> produced is

$$|\psi> = A|0> \equiv |\psi_1> + |\psi_0> , \quad (36)$$

where $|\psi_i> \in \mathcal{H}_i$ {i = 0,1}. Defining $|n_i> = |\psi_i>/\sqrt{<\psi_i|\psi_i>}$ and $a = <\psi_1|\psi_1>$, Eq. (36) becomes

$$|\psi> = \sqrt{a}\,|n_1> + \sqrt{1-a}\,|n_0> . \quad (37)$$

Here $\sqrt{a}$ is the quantum amplitude that a measurement in the CB made at the end of $\mathcal{A}$ will yield a good state. This is also the setting for the QAEA whose task is to return an estimate of $\sqrt{a}$.

Central to quantum amplitude amplification is a Grover-inspired operator Q whose action on the subspace $\mathcal{H}_\psi$ spanned by $|n_1>$ and $|n_0>$ is given by:

$$Q = U_\psi U_{n_0}\,, \tag{38}$$

where $U_\psi = I - 2|\psi><\psi|$ and $U_{n_0} = I - 2|n_0><n_0|$. The U operators are clearly unitary and so then is Q. We explain below how the action of Q is extended to the orthogonal complement $\mathcal{H}_\psi^\perp$ of $\mathcal{H}_\psi$ (in $\mathcal{H}$).

Since |ψ> is normalized, it is possible to write $\sqrt{a} = \sin\theta_a$, with $0 \le \theta_a \le \pi/2$. Straight-forward algebra shows that Q implements a rotation of the subspace $\mathcal{H}_\psi$:

$$Q|n_1> = \cos 2\theta_a |n_1> - \sin 2\theta_a\ |n_0> \tag{39a}$$

$$Q|n_0> = \sin 2\theta_a\ |n_1> + \cos 2\theta_a\, |n_1>\,. \tag{39b}$$

To fully specify the action of Q on $\mathcal{H}$ it is necessary to determine its action on the orthogonal complement $\mathcal{H}_\psi^\perp$. To that end, an alternative form for Q is used[12]:

$$Q = -AS_0A^{-1}S_\chi\,, \tag{40}$$

where the action of S_0 on the CB states is

$$S_0|j> = \begin{cases} |j> & (j \neq 0) \\ -|0>\,, \end{cases} \tag{41}$$

and that of S_χ is

$$S_\chi\,|j> = (-1)^{\chi(j)}\,|j>\,. \tag{42}$$

Thus S_χ changes the sign of the good CB states and fixes the bad ones.

From Eq. (41) and the definition of |ψ> we see that

$$AS_0A^{-1} = I - 2|\psi><\psi|.$$

Since $|\psi> \in \mathcal{H}_\psi$, it is orthogonal to all states in $\mathcal{H}_\psi^\perp$, and so AS_0A^{-1} acts as the identity on $\mathcal{H}_\psi^\perp$.

Recall that the CB states partition into good and bad states which will be denoted by $\{|j_g>\}$ and $\{|j_b>\}$, respectively. It is possible to form an alternative orthonormal basis for $\mathcal{H}_1$ that includes $|n_1>$, and one for $\mathcal{H}_0$ that includes $|n_0>$, which we denote, respectively, by $\{|n_1>, \{|n_\gamma^g>\}\}$ and $\{|n_0>,\ \{|n_\beta^b>\}\}$. Thus, $|n_1>$ and $|n_0>$ span $\mathcal{H}_\psi$ (as before), and the $\{|n_\gamma^g>, |n_\beta^b>\}$ span $\mathcal{H}_\psi^\perp$. By construction, each $|n_\gamma^g>$ is a linear combination of the $\{|j_g>\}$ orthogonal to $|n_1>$, and each $|n_\beta^b>$ is a linear combination of the $\{|j_b>\}$ orthogonal to $|n_0>$. It follows from this and Eq. (42) that

$$S_\chi\,|n_\gamma^g> = -|n_\gamma^g> \tag{43a}$$

$$S_\chi \, |n_\beta^b> = \; |n_\beta^b> \; , \tag{43b}$$

and similarly, for linear combinations of good (bad) basis vectors $\{|n_\gamma^g>\}$ ($\{|n_\beta^b>\}$). Thus, for a state $|\phi> = |\phi_g> + |\phi_b> \in \mathcal{H}_\psi^\perp$, Eqs. (40) and (43) give

$$Q|\phi> = |\phi_g> - |\phi_b> \tag{44}$$

and consequently

$$Q^2 \, |\phi> = |\phi> \, . \tag{45}$$

Thus, Q flips the sign of the bad component of $|\phi>$ and fixes the good component, while Q^2 acts as the identity on $\mathcal{H}_\psi^\perp$.

It proves useful to write out the eigenstates and eigenvalues of Q. The non-trivial case assumes $a \neq 0, 1$ (viz. $\theta_a \neq 0, \pi/2$). Then $\mathcal{H}_\psi$ is two-dimensional, and a simple calculation shows that the eigenstates of Q in $\mathcal{H}_\psi$ are

$$|\psi_\pm> = \frac{1}{\sqrt{2}} [|n_1> \pm i \, |n_0>] \, , \tag{46}$$

with eigenvalues

$$\lambda_\pm = \; e^{\pm 2i\theta_a} \; . \tag{47}$$

For states in $\mathcal{H}_\psi^\perp$, we saw that $Q^2 = I$. Thus, for this subspace: (i) the eigenvalues of Q are $\lambda = \pm 1$; and (ii) from Eqs. (40) and (43), the +1 (-1) eigenstates are the elements of the good (bad) subspace of $\mathcal{H}_\psi^\perp$.

Eqs. (46) can be inverted to give

$$|n_1> = \; \frac{1}{\sqrt{2}} \, [|\psi_+> + |\psi_->]$$

$$|n_0> = \; \frac{-i}{\sqrt{2}} \, [|\psi_+> - |\psi_->] \, .$$

Eq. (37) then becomes

$$|\psi> = \; A|0> = \; \frac{-i}{\sqrt{2}} \left[e^{i\theta_a} \, |\psi_+> - \; e^{-i\theta_a} \, |\psi_->\right] , \tag{48}$$

and it follows from Eqs. (46) and (47) that

$$Q^j \, |\psi> = \; \sin[(2j+1)\theta_a] \, |n_1> \; + \; \cos[(2j+1)\theta_a] \, |n_0> \; . \tag{49}$$

This establishes the basic results needed to understand the construction of the QAEA. Before beginning that discussion in the following subsection, we point out how the quantum amplitude

amplification algorithm follows from the above. In the case of most interest, $\theta_a \ll 1$ (viz. $\sqrt{a} \ll 1$). If we prepare the state |ψ> as above using the quantum algorithm $\mathcal{A}$ and then measure it in the CB, the probability of getting a good state is $a = sin^2\theta_a \ll 1$. If, however, we apply Q^j to |ψ> and then measure in the CB, we see from Eq. (48) that the probability to obtain a good state is now $sin^2[(2j+1)\theta_a]$. If j is chosen such that $(2j+1)\theta_a \approx \pi/2$, we produce a good state with probability near 1, thus greatly amplifying the success probability. This, then, is the quantum amplitude amplification algorithm which can be shown to succeed even when θ_a is unknown (see Theorem 3 in Ref. 12). It is important not to confuse the quantum amplitude amplification algorithm with the QAEA. The former produces a good quantum state with high probability, while the later gives an estimate of the unknown quantum amplitude $\sqrt{a}$.

SI-6B Quantum amplitude estimation algorithm

As noted at the end of Section SI-6A, the QAEA returns an estimate of the quantum amplitude $\sqrt{a} = \sin\theta_a$, or equivalently, of the probability $a = sin^2\theta_a$ to obtain a good quantum state when measuring the state $|\psi> = A|0>$ in the CB. As will be seen below, it does this by using a slight modification of the quantum phase estimation algorithm[16] to return an estimate of θ_a, and hence of $\sqrt{a}$ and a.

Let $\mathcal{H}$ denote the Hilbert space of an n-qubit register of dimension $N = 2^n$. To set notation we make three definitions. (1) We begin with the well-known quantum Fourier transform (QFT) F_N whose action on the CB states {|j>} of $\mathcal{H}$ is:

$$F_N\,|j> = \frac{1}{\sqrt{N}} \sum_{k=0}^{N-1} e^{2\pi i\, jk/N}\,|k>\,.$$

(2) As will be seen shortly, the QAEA works with a system of two n-qubit registers whose Hilbert space $\widetilde{\mathcal{H}} = \mathcal{H}\otimes\mathcal{H}$ has CB states $|j> \otimes |k>$, with $j,k \in \{0,1,\ldots,N-1\}$. The algorithm makes use of an operator $\Lambda_N(Q)$ which has the following action on the CB states:

$$\Lambda_N(Q)|j> \otimes |k> = \big|j> \otimes Q^j\big|k>\,.$$

(3) Finally, let $0 \le \omega < 1$. It proves useful to introduce the state $|S_N(\omega)>$, where

$$|S_N(\omega)> = \frac{1}{\sqrt{N}} \sum_{k=0}^{N-1} e^{2\pi i\,\omega k}\,|k>\,.$$

Note that for $j \in \{0,1,\ldots,N-1\}$, we have $|S_N(j/N)> = F_N|j>$.

The QAEA[12] consists of the following 6 steps – the notation follows Section SI-6A:

1. Initialize the two n-qubit registers to the state $|0> \otimes A|0>$.
2. Apply the QFT F_N to the first register.
3. Apply $\Lambda_N(Q)$ to the two n-qubit register system.

4. Apply the inverse QFT F_N^{-1} to the first register.
5. Measure the first register in the CB without observing the second register. Let *y* denote the measurement result.
6. Return $\tilde{\theta}_a = \pi y/N$ as the estimate for θ_a, and hence $\tilde{a} = sin^2\tilde{\theta}_a$ as the estimate for *a*.

Steps 1-5 implement the quantum phase estimation algorithm[16] for Q with the slight modification that the initial state A|0> of the second register is a superposition of eigenstates of Q (see Eq. (48) above). The performance of the QAEA is given by Theorem 12 in Ref. 12 which is reproduced here for the reader's convenience. The prerequisites for the QAEA are the operator A, the Boolean function χ, and the dimension N of the Hilbert space $\mathcal{H}$.

Theorem 12: *For any positive integer k, the QAEA returns the estimate* $\tilde{a}$ $(0 \leq \tilde{a} \leq 1)$ *for the probability* $a = <\psi_1|\psi_1>$ *such that the error* $|\tilde{a} - a|$ *satisfies the upper bound*

$$|\tilde{a} - a| \leq 2\left(\frac{k\pi}{N}\right)\sqrt{a(1-a)} + \left(\frac{k\pi}{N}\right)^2 ,$$

with probability at least $8/\pi^2$ *when k = 1, and with probability greater than* $1 - 1/[2(k-1)]$ *for* $k \geq 2$. *The algorithm uses exactly N evaluations of χ. If a = 0, then* $\tilde{a} = 0$ *with certainty, and if a = 1 and N is even, then* $\tilde{a} = 1$ *with certainty.*

Remarks: (i) Since we have assumed that $N = 2^n$, the QAEA returns $\tilde{a} = 1$ with certainty when $a = 1$. (ii) Since $\sqrt{a(1-a)} \leq 1/2$, it follows that for $N \gg 1$, the error $|\tilde{a} - a| \sim k\pi/N$. By choosing *N* large enough, we can make this error as small as we like. Notice that it vanishes exponentially with the number of qubits n.

We illustrate the use of the QAEA in the following section.

SI-6C Estimating the mean-value of a function

Following Novak[15], we explain here how the QAEA can be used to estimate the mean-value of a function. This is the *only task* in our quantum PDE algorithm that requires a quantum computer. As in Section SI-4, we assume that the function *f* is a real-valued Hölder class function. As shown in the paper accompanying this SI, *f* can be shifted and rescaled so that the resulting function *g* takes values in the interval [0,1]. The QAEA is then used to estimate the mean-value of *g*. By reversing the shift and rescaling it is possible to determine the mean-value of *f* from the mean-value of *g*. Throughout this Section then, we assume $0 \leq g(t) \leq 1$.

Suppose that *g* is given at *N* instants of time, $g(t_j) \equiv g(j)$, with $j \in \{0, \ldots, N-1\}$. The task is to estimate the mean-value

$$\bar{g} = \frac{1}{N}\sum_{j=0}^{N-1} g(j) .$$

Note that $0 \leq \bar{g} \leq 1$ since $0 \leq g(j) \leq 1$. As *g* is now a real-valued function and not a Boolean function, a number of modifications must be introduced to the discussion given in

Sections SI-6A and SI-6B. (1) First, an (n+1)-qubit Hilbert space $\mathcal{H}' = \mathcal{H}\otimes\mathcal{H}_c$ is introduced, where $\mathcal{H}$ is an n-qubit Hilbert space with CB states $\{|j>: j \in \{0, \dots, N-1\}\}$ and $\mathcal{H}_c$ is a one-qubit Hilbert space with CB states $|0>_1$ and $|1>_1$. The subscript on the $\mathcal{H}_c$ CB states is meant to remind us that these are one-qubit states. The absence of such a subscript on a ket indicates that it is an n-qubit state. We will refer to the qubit associated with $\mathcal{H}_c$ as the chaperon qubit as it will be seen to monitor which (n+1)-qubit states are good states and which are bad. The CB states for $\mathcal{H}'$ are $|j>\otimes|0>_1$ and $|j>\otimes|1>_1$, with $j \in \{0, \dots, N-1\}$. (2) Novak introduces a quantum oracle $\mathcal{O}$ whose action on the CB states is:

$$\mathcal{O}|j>\otimes|1>_1 = \sqrt{g(j)}\,|j>\otimes|1>_1 + \sqrt{1-g(j)}\,|j>\otimes|0>_1 \qquad (50a)$$

$$\mathcal{O}|j>\otimes|0>_1 = -\sqrt{1-g(j)}\,|j>\otimes|1>_1 + \sqrt{g(j)}\,|j>\otimes|0>_1 \ . \qquad (50b)$$

The oracle $\mathcal{O}$ thus applies a unitary operation that encodes the N values of g into superpositions of the CB states. (3) The operator A appearing in Sections SI-6A and SI-6B is here defined to be:

$$A = \mathcal{O}\,(F_N\otimes I_c)\,.$$

It applies the QFT F_N to the n non-chaperon qubits and the identity operation to the chaperon qubit. It then applies the oracle $\mathcal{O}$ to the (n+1)-qubit system. If the (n+1)-qubit state is $|0>\otimes|1>_1$, A maps it to the state $|\psi>$:

$$|\psi> = A|0>\otimes|1>_1 = \frac{1}{\sqrt{N}}\sum_{j=0}^{N-1}\left[\sqrt{g(j)}\,|j>\otimes|1>_1 + \sqrt{1-g(j)}\,|j>\otimes|0>_1\right] . \qquad (51)$$

Introducing the normalized states

$$|n_1> = \frac{1}{\sqrt{N}}\sum_{j=0}^{N-1}\sqrt{\frac{g(j)}{\bar{g}}}\,|j>\otimes|1>_1 \qquad (52a)$$

$$|n_0> = \frac{1}{\sqrt{N}}\sum_{j=0}^{N-1}\sqrt{\frac{1-g(j)}{\bar{g}}}\,|j>\otimes|0>_1 \ , \qquad (52b)$$

where $\bar{g}$ is the desired mean-value, Eq. (51) becomes

$$|\psi> = A|0>\otimes|1>_1 = \sqrt{a}\,|n_1> + \sqrt{1-a}\,|n_0> \ . \qquad (53)$$

Here $a = \bar{g}$, which follows from Eqs. (51) and (52). As in Sections SI-6A and SI-6B, we can write $a = sin^2\theta_a$, with $0 \le \theta_a \le \pi/2$. Obtaining an estimate of θ_a then gives the desired estimate of $\bar{g}$.

Novak's construction has thus produced a state $|\psi>$ which has the same form as Eq. (37) above. It is possible to repeat the construction of the Grover operator Q given in Section SI-6A using the states $|n_1>$ and $|n_0>$ appearing in Eq. (52). As before, these states span the subspace $\mathcal{H}'_\psi$, with $|n_1>$ the good state and $|n_0>$ the bad state. The good and bad basis states for the orthogonal

complement $\mathcal{H}'^{\perp}_{\psi}$ are similarly constructed, respectively, $\{|n^g_\gamma> \otimes |1>_1\}$ and $\{|n^b_\beta> \otimes |0>_1\}$. By construction, the good (bad) states have the chaperon qubit in the state $|1>_1$ ($|0>_1$). The operator Q implements the same operations on the good and bad states as in Section SI-6A. This allows Eq. (51) to be rewritten as

$$|\psi> = \; A|0> \otimes |1>_1 = \; -\frac{i}{\sqrt{2}} \left[e^{i\theta_a} \, |\psi_+> \; - \; e^{-i\theta_a} \, |\psi_->\right] , \qquad (54)$$

where $|\psi_\pm>$ are the eigenstates of Q in $\mathcal{H}'_\psi$,

$$Q|\psi_\pm> = \; e^{\pm 2i\theta_a} \, |\psi_\pm> \; ,$$

which are given as in Eq. (46), though using the $|n_1>$ and $|n_0>$ appearing in Eq. (52).

We now have the necessary results in hand to estimate the mean-value $\bar{g} = a = \; sin^2\theta_a$ using the QAEA. To begin, we enlarge our system by adding another n qubits so that the full Hilbert space is $\widetilde{\mathcal{H}} \; = \; \mathcal{H} \otimes \mathcal{H}'$. We refer to the n-qubits associated with $\mathcal{H}$ as the first register, and the n+1 qubits associated with $\mathcal{H}'$ as the second register. The CB states for $\widetilde{\mathcal{H}}$ are $|j> \otimes (|k> \otimes |0>_1)$ and $|j> \otimes (|k> \otimes |1>_1)$, with $j, k \; \in \; \{0, \dots, N-1\}$. We now walk through the 6 steps of the QAEA given in Section SI-6B to obtain the estimate $\tilde{\bar{g}}$ of the mean-value $\bar{g}$:

1. Apply $I_n \otimes A$ to the initial state $|0> \otimes (|0> \otimes |1>_1)$. This gives the state
$$|0> \otimes |\psi> \; = \; \; |0> \otimes (A|0> \otimes |1>_1) \, .$$
2. Apply the QFT F_N to the first register, giving the state
$$-\frac{i}{\sqrt{2N}} \sum_{j=0}^{N-1} |j> \otimes \left[e^{i\theta_a} \, |\psi_+> \; - \; e^{-i\theta_a} \, |\psi_->\right] \, .$$
3. Apply $\Lambda_{2N^2}(Q)$ to the two registers. The resulting state is
$$\frac{e^{i\theta_a}}{\sqrt{2}} \, |S_N\left(\frac{\theta_a}{\pi}\right)> \otimes |\psi_+> \; - \; \frac{e^{-i\theta_a}}{\sqrt{2}} \, |S_N\left(1 - \frac{\theta_a}{\pi}\right)> \otimes |\psi_-> \; .$$
4. Apply the inverse QFT F_N^{-1} to the first register giving
$$\frac{e^{i\theta_a}}{\sqrt{2}} \, F_N^{-1} \, |S_N\left(\frac{\theta_a}{\pi}\right)> \otimes |\psi_+> \; - \; \frac{e^{-i\theta_a}}{\sqrt{2}} \, F_N^{-1} \, |S_N\left(1 - \frac{\theta_a}{\pi}\right)> \otimes |\psi_-> \; .$$
5. Measure the first register without observing the second register. This leaves the first register in the mixed state
$$\frac{1}{2} \, F_N^{-1} \left|S_N\left(\frac{\theta_a}{\pi}\right)><S_N\left(\frac{\theta_a}{\pi}\right)\right| \, F_N \; + \; \frac{1}{2} \, F_N^{-1} \left|S_N\left(1 - \frac{\theta_a}{\pi}\right)><S_N\left(1 - \frac{\theta_a}{\pi}\right)\right| \, F_N \; . \quad (55)$$
Suppose the measurement outcome is $y \; \in \; \{0, \dots, N-1\}$. With probability $1/2$, the first register is in the state given by the first term in Eq. (55). As shown in Ref. 12, in this case, the probability distribution for y/N is sharply-peaked about θ_a/π for N large. Similarly, with probability $1/2$, the first register is in the state given by the second term in Eq. (55). Here the probability distribution is sharply-peaked about $(\pi - \theta_a)/\pi$.

6. In the first case discussed in Step 5, the QAEA returns the estimate $\tilde{\theta}_{a,1} = \pi y/N$ and so $\tilde{a}_1 = sin^2(\pi y/N)$. In the second case, the QAEA returns $\tilde{\theta}_{a,2} = \pi(N-y)/N$ and $\tilde{a}_2 = sin^2(\pi(N-y)/N)$. Notice that $\tilde{a}_1 = \tilde{a}_2$ so that the final estimate for $\tilde{a}$ is

$$\tilde{a} = \frac{1}{2}\tilde{a}_1 + \frac{1}{2}\tilde{a}_2 = sin^2\left(\frac{\pi y}{N}\right) .$$

This gives the estimated mean-value

$$\tilde{\bar{g}} = sin^2\left(\frac{\pi y}{N}\right) ,$$

where y is the measurement outcome in Step 5.

Theorem 12 in Ref. 12 (see also Section SI-6B) gives an upper bound for the error $\delta\bar{g} = |\tilde{a} - a|$. For $N \gg 1$ and $k = 1$, the upper bound reduces to

$$\delta\bar{g} \leq \frac{\pi}{N} ,$$

which is satisfied with probability greater than $8/\pi^2$. If we choose $N > \pi/\epsilon$, then

$$\delta\bar{g} < \epsilon .$$

We can thus make the error $\delta\bar{g}$ as small as we like by choosing N sufficiently large.

References

1. Anderson, J. D. *Fundamentals of Aerodynamics 6th ed.* (McGraw Hill, 2017).
2. Anderson, J. D. *Modern Compressible Flow 3rd ed.* (McGraw Hill, 2003).
3. Pletcher, R. H. Tannehill, J. C. & Anderson, D. A. *Computational Fluid Mechanics and Heat Transfer, 3rd ed.* (CRC Press, 2013).
4. Fletcher, C. A. J. *Computational Techniques for Fluid Dynamics 2nd ed., Vol. 1* (Springer, 1991).
5. Anderson, J. D. *Computational Fluid Dynamics* (McGraw Hill, 1995).
6. Kacewicz, B. Almost optimal solution of initial-value problems by randomized and quantum algorithms. *J. Complexity* **22**, 676-690 (2006).
7. Evans, L. C. *Partial Differential Equations*. (American Mathematical Society, 1998).
8. Gilbarg, D. & Trudinger, N. *Elliptic Partial Differential Equations of Second Order*. (Springer, 1983).
9. Iserles, A. *A First Course in the Numerical Analysis of Differential Equations*. (Cambridge, 2009).
10. Atkinson, K. *Elementary Numerical Analysis*. (Wiley, 1985).
11. Moursund, D. G. & Duris, C. S. *Elementary Theory and Application of Numerical Analysis*. (Dover, 1988).
12. Brassard, G. Hoyer, P. Mosca, M., Tapp, A. Quantum amplitude amplification and estimation. Preprint at http://arXiv.org/quant-ph/0005055 (2000).
13. Heinrich, S. Quantum summation with an application to integration. *J. Complexity* **18**, 1-50 (2002), see Lemma 3.

14. Courant, R. Friedrichs, K. O. Lewy, H. Über die Partiellen Differenzengleichungen der Mathematischen Physik. *Math. Ann.* **100**, 32-74 (1928). (Translation: On the partial difference equations of mathematical physics. *IBM J. Res. Dev.* **11**, 215-234 (1967).)
15. Novak, E. Quantum complexity of integration. *J.* Complexity **17**, 2-16 (2001).
16. Kitaev, A. Quantum measurements and the Abelian stabilizer problem. Preprint at http://arXiv.org/quant-ph/9511026 (1995).
17. Grover, L. K. Quantum mechanics helps in searching for a needle in a haystack. Phys. Rev. Lett. **79**, 325-328 (1997).
18. Kacewicz, B. Randomized and quantum algorithms yield a speed-up for initial value problems. *J. Complexity* **20**, 821-834 (2004).
19. Kacewicz, B. Improved bounds on randomized and quantum complexity of initial-value problems. *J. Complexity* **21**, 740-756 (2005).