



A conceptual framework for quantum accelerated automated design optimization

Matthias Möller^{a,*}, Cornelis Vuik^a

Delft University of Technology, Delft Institute of Applied Mathematics (DIAM), Van Mourik Broekmanweg 6, XE Delft 2628, The Netherlands

ARTICLE INFO

Article history:

Received 20 June 2018

Revised 6 January 2019

Accepted 12 February 2019

Available online 13 February 2019

Keywords:

Automated design optimization

High-performance computing

Quantum algorithms

Quantum-accelerated computing

Scientific computing

ABSTRACT

The development of practical quantum computers that can be used to solve real-world problems is in full swing driven by the ambitious expectation that quantum supremacy will be able to outperform classical super-computers. Like with any emerging compute technology, it needs early adopters in the scientific computing community to identify problems of practical interest that are suitable as proof-of-concept applications and to revise existing solution strategies and develop new ones that exploit the capabilities of the novel compute hardware.

In this article we describe a conceptual framework for reducing the computational complexity of simulation-driven automated design optimization processes, which are nowadays widely used in computer-aided product development, by exploiting quantum supremacy. Our approach is based on the assumption that quantum computers will become part of hybrid high-performance computing platforms and can then be used as application-specific accelerator devices.

© 2019 Elsevier B.V. All rights reserved.

1. Introduction

The *era of accelerated computing* started in the mid-2000s, when CPU clock speeds approached the 4GHz barrier and a further increase beyond this barrier would have required enormous efforts for cooling the processor to prevent spurious malfunctioning and even permanent hardware damage from overheating. All major chip vendors followed the paradigm shift from chasing ultimate single-core performance towards developing parallel high-performance computing (HPC) technologies and flooded the market with multi-core CPUs and many-core accelerator cards like programmable GPUs and dedicated co-processor devices.

1.1. Accelerated computing

The key idea of accelerated computing is to offload computationally expensive tasks from the host, a classical multi-core and possibly multi-socket CPU-based computer, to the attached accelerator devices, which altogether form the so-called compute node. Modern HPC systems consist of hundreds and thousands of compute nodes, which are interconnected by high-speed networks.

In classical accelerated computing, the role of the host computer is threefold: Firstly, tasks that do not benefit from the compute capabilities of the accelerator devices such as in- and output of data from and to the global filesystem and intrinsically sequential (parts of) algorithms are executed by the host. Secondly, the host is responsible for orchestrating the interplay of accelerator devices among each other and with the CPU and for managing the communication between the distributed compute nodes. Finally, since modern CPUs have up to 20–32 cores with integrated vector-processing units, heterogeneous HPC systems also use the massive compute power of the host to perform actual computations.

Most of today's many-core accelerators are designed for executing parallelizable and/or vectorizable instructions of SIMD-type (single instruction multiple data) exceptionally fast. Consider, for instance, the multiplication of an $m \times n$ matrix with a column vector of length n . Each matrix row gives rise to a separate dot product, i.e. an accumulated multiply-add operation that can be carried out in a parallel and, ideally, vectorized loop over all rows even on multiple devices with distributed memory architecture. This so-called divide-and-conquer approach is a common building block in classical HPC applications and it is supported by most programming models like OpenMP [1] and MPI [2].

Recently, application-specific accelerator technologies are emerging, which offer extra functionality that is not available in commodity hardware. Consider, for instance, Google's tensor processing units [3], which is an application-specific integrated

* Corresponding author.

E-mail address: m.moller@tudelft.nl (M. Möller).

URL: <http://www.ta.twi.tudelft.nl/nw/users/matthias/> (M. Möller)

circuit (ASIC) for accelerating machine learning applications. To fully exploit its compute power, the part of the application that benefits from using the AI accelerator needs to be identified and implemented in the vendor-specific programming model, which often requires code refactoring.

1.2. Quantum-accelerated computing

In our opinion, quantum computing has the potential of becoming a disruptive application-specific acceleration technology that might have a significant impact on future developments in high-performance scientific computing [4]. However, this variant of accelerated computing is so much different from existing technologies, that it needs radically new algorithmic concepts rather than the continuation and extension of traditional approaches to achieve quantum supremacy, that is, the potential ability of quantum computing devices to solve problems that cannot be solved efficiently by classical computers [5].

Consider, for instance, the aforementioned matrix-vector multiplication that can easily be accelerated in classical computing by adopting the divide-and-conquer approach. The very limited number of qubits in today's (50-qubit processor [6]) and mid-future (72- and 128-qubit processors [7,8]) quantum devices makes this strategy of parallelizing along the problem size unattractive.

Another concept, which is widely used for solving boundary value problems (BVP) that are modeled by partial differential equations (PDE), are so-called domain decomposition methods (DDM) [9]. The key idea is to split a single large problem into multiple smaller ones that can be solved in parallel on multiple distributed compute devices. Data is regularly exchanged between the different sub-domains to ensure that the global solution that is made up from local parts solves the original problem. The straightforward application of DDMs on quantum devices is ruled out by the no-cloning theorem [10], which states that it is impossible to create an identical copy of an arbitrary unknown quantum state. Of course, measurements could be performed in order to exchange classical state data between sub-domains but that will most likely stop quantum supremacy.

It should be noted that direct communication between quantum devices is possible without destroying the superposition of quantum states via quantum teleportation [11] and quantum channels [12], but this requires a conceptual redesign of the DDM, which typically performs simple averaging of the non-unique data available at the sub-domain interfaces. The above is not meant to discourage practitioners from looking into quantum-accelerated computing but aims to identify some of the many challenges that one might encounter.

In what follows, we sketch a conceptual framework for accelerating the solution of simulation-based automated design optimization (ADO) problems with the aid of quantum devices. Our theoretical scenario is based on existing quantum algorithms for solving linear systems of equations and for finding the minimum of a quadratic form. The rest of this paper is organized as follows: Section 2 establishes an abstract mathematical framework for solving ADO problems and gives two illustrative examples. Section 3 reviews existing quantum algorithms that can be used to realize the suggested ADO framework in practice. Finally, a brief discussion of remaining open questions, recommendations for further research and main conclusions are given in Section 4.

2. Automatic design optimization

Computer-aided product development is a key technology of Industry 4.0. Consider the design of a full airplane or parts of it like its engines, which have reached a complexity level that renders the manual adjustment of the hundreds and thousands of individual

design parameters impractical. A remedy is to use automated design optimization (ADO) procedures, in which the parameters are varied systematically based on computer simulations that provide a prediction of the flow field and other physical processes that need to be taken into account.

Simply speaking, the task is to find, say, a wing design with minimal weight, maximal durability and resilience, and beneficial aerodynamic shape that ensures safe maneuvering of the airplane in all design and a wide range of off-design conditions, which moreover reduces CO₂, NO_x and noise emissions and satisfies international regulations concerning, say, the maximal wingspan.

It should be clear that most problems of practical interest do not possess a unique optimal solution. In this case, an 'optimal' design is only the best possible compromise between the many and often conflicting requirements.

In what follows, we derive an abstract ADO framework using the example of optimal wing design for illustration purposes. A model problem that meets all prerequisites of the quantum algorithms from Section 3 is introduced afterwards.

2.1. PDE-Constrained optimization

ADO problems can be formulated as continuous minimization problems that are constrained by one or more PDEs and additional equality and inequality constraints. To begin with, let $\alpha = (\alpha_1, \dots, \alpha_M)^T$ denote the vector of design variables, which need to be chosen from the 'admissible' design space \mathcal{D} which forms a subset of the \mathbb{R}^M and is thus very difficult to explore in full detail when $M \gg 1$. This challenge is widely known as the 'curse of dimensionality'.

In practice, the admissibility conditions are of the form

$$\alpha_i^{\min} \leq \alpha_i \leq \alpha_i^{\max}, \quad i = 1, \dots, M \quad (1)$$

combined with more complicated equality and inequality constraints, e.g.,

$$L_j < \mathbf{c}_j^T \alpha \leq U_j, \quad j = 1, \dots, N, \quad (2)$$

where the dot product $\mathbf{c}_j^T \alpha$ stands for a linear combination of the design variables and L_j and U_j are lower and upper bounds, respectively.

As an example, let α_1 be the wingspan, which must satisfy international standards, e.g., $52\text{m} \leq \alpha_1 < 65\text{m}$ to satisfy the International Civil Aviation Organization (ICAO) Code E requirements. Moreover, the expected fuel capacity of the wing tanks sets up a lower bound for the minimal wing volume, which yields a constraint of the second type.

Each α generates a particular design, termed the control $U(\alpha)$, which describes the shape of the wing in the first place but can include other properties as well. Its quality is assessed with respect to key performance indicators. A crucial quantity in aerodynamics is the lift-to-drag coefficient c_l/c_d that should be sufficiently high in order to obtain designs with good aerodynamic properties. Drag and lift coefficients are computed from the flow velocity \mathbf{v} and pressure p , which can be calculated, e.g., by the Navier-Stokes equations or another so-called flow model. Such flow models typically consist of a set of (coupled) PDEs, which are complemented by consistent boundary and initial conditions.

Without going into the mathematical details, let

$$\mathcal{R}(U(\alpha); Y) = 0 \quad (3)$$

denote the flow model in abstract residual form. That is, given a control $U(\alpha)$, our aim is to find a solution $Y = (\mathbf{v}, p)$ that satisfies the flow model (3). It goes without saying that the solution depends on the control, i.e. $Y = Y(U(\alpha))$.

Next, let us define the cost functional that should be minimized

$$\mathcal{J}(U(\boldsymbol{\alpha}); Y) = \left(\frac{c_l}{c_d} \right)^{-1}. \quad (4)$$

Then the task of creating an *optimal* wing design $U(\boldsymbol{\alpha}^*)$ amounts to finding an admissible configuration $\boldsymbol{\alpha}^* \in \mathcal{D}$ such that the solution $Y(U(\boldsymbol{\alpha}^*))$ to the flow model (3) also minimizes the cost functional (4), i.e.

$$\min_{\boldsymbol{\alpha} \in \mathcal{D}} \mathcal{J}(U(\boldsymbol{\alpha}); Y), \quad (5)$$

$$\text{s.t. } \mathcal{R}(U(\boldsymbol{\alpha}); Y) = 0. \quad (6)$$

In truly *multi-disciplinary* and *multi-objective* design optimization (MDO) problems the aim is to minimize several cost functionals simultaneously, whereby the solution Y needs to satisfy several PDE problems, e.g., the Navier-Stokes equations for predicting the behavior of the flow around the wing and another structural mechanics model for estimating the stresses acting inside it.

A common strategy for solving the ADO problem (5)–(6) is to evaluate the PDE problem for many different parameter sets $\{\boldsymbol{\alpha}\}_g$ in parallel, select a few best ones with respect to the cost functional \mathcal{J} , combine them using nature-inspired mutation, crossover, and selection procedures and reevaluate (6) with the next generation of design parameters $\{\boldsymbol{\alpha}\}_{g+1}$. This procedure is repeated many times until a sufficiently good design has been found. The main bottleneck of this evolutionary or genetic algorithm approach [13] are the huge computational costs due to the frequent evaluation of the PDE problem, which can take days or even weeks for a single realistic simulation run.

Next to such meta-heuristic search strategies, which might not lead to an immediate design improvement in each single optimization cycle, there also exist gradient-based optimization algorithms [14], which compute the gradient of the cost functional with respect to the design parameters and select the next configuration of design parameters in such a way that \mathcal{J} improves the most. However, these methods run the risk of being trapped in a local (but not global) minimum, from which a gradient-based method cannot escape by design.

A more recent approach is to combine gradient-free and gradient-based strategies into so-called hybrid optimization algorithm [15,16], which combine the computational efficiency of gradient-based methods with the strength of gradient-free approaches to explore the full search space.

2.2. Academic model problem

For the further discussion, let us consider a much simpler model problem that fits into the abstract ADO framework but allows us to easily verify the prerequisites of the quantum algorithms to be addressed in Section 3.

Consider Poisson's equation in two spatial dimension with homogeneous Dirichlet boundary conditions prescribed along the entire boundary $\Gamma = \partial\Omega$:

$$\Delta u + f = 0 \quad \text{in } \Omega, \quad (7)$$

$$u = 0 \quad \text{on } \Gamma. \quad (8)$$

The domain $\Omega = \Omega(\alpha)$ is given by the unit square, whereby the shape of the lower boundary part can be varied within the range of the quadratic polynomial

$$y(x) = \alpha(x - x^2), \quad \alpha^{\min} \leq \alpha \leq \alpha^{\max}, \quad 0 \leq x \leq 1. \quad (9)$$

The aim is to minimize the L_2 -error between the solution u to the above BVP (7)–(8) and a given reference profile u^* prescribed within $\Omega(\alpha)$, i.e.

$$\mathcal{J}(\Omega(\alpha); u) = \sqrt{\int_{\Omega(\alpha)} (u - u^*)^\top (u - u^*) \, \mathbf{d}\mathbf{x}}. \quad (10)$$

Without going into the mathematical details, we remark that the BVP (7)–(8) is typically approximated by a numerical discretization scheme, like the finite difference method (FDM) [17], the finite volume method (FVM) [18] or the finite element method (FEM) [19], which leads to the linear system of equations

$$A_h u_h = f_h, \quad (11)$$

where the stiffness matrix $A_h \approx \Delta(\cdot)$ represents the discretized differential operator, and $u_h \approx u$ and $f_h \approx f$ are approximations of the unknown solution and the load vector, respectively. Let us approximate the target profile $u_h^* \approx u^*$ and define the auxiliary vectors $y_h = u_h - u_h^*$, and $b_h = f_h - A_h u_h^*$. By subtracting $A_h u_h^*$ from both sides of (11), we arrive at the equivalent formulation

$$A_h y_h = b_h, \quad (12)$$

whose solution y_h can be directly inserted into the cost functional

$$\mathcal{J}(\Omega_h(\alpha); y_h) = \sqrt{\int_{\Omega_h(\alpha)} y_h^\top y_h \, \mathbf{d}\mathbf{x}}, \quad (13)$$

where $\Omega_h(\alpha) \approx \Omega(\alpha)$ is the approximation of the domain by the mesh. The integral term is typically evaluated by numerical quadrature leading to

$$j_h = \sqrt{y_h^\top M_h y_h} = \sqrt{\langle y_h | M_h | y_h \rangle}, \quad (14)$$

where M_h is the consistent mass matrix for the FEM and the identity matrix for FDM and FVM. Both A_h and M_h are symmetric $N \times N$ matrices, where N is the number of entries in vector y_h . For most spatial discretization schemes and, in particular, the FDM, FVM, and FEM, the matrices are s -sparse, that is, each row contains at most $s \ll N$ non-zero entries. A_h is moreover symmetric positive-definite and efficiently row computable, that is, each entry A_{ij} can be accessed in $\mathcal{O}(s)$ time from the index pair (i, j) if the matrix entries are stored in compressed sparse row (CSR) format [20]. Without the capability of storing A_h explicitly each matrix entry can be computed on-the-fly in $\mathcal{O}(d)$ time if Lagrange finite element basis functions are adopted on d -simplexes [21].

The condition number $\kappa(A_h) = \|A_h\| \|A_h^{-1}\|$ is proportional to $\mathcal{O}(h_{\min}^{-2})$, where h_{\min} is the minimal element length [22]. However, the effective condition number of A_h is typically much smaller [23], which is particularly helpful to accelerate the solution of the linear system of Eqs. (12) and to solve the overall optimization problem (5)–(6) using efficient quantum algorithms.

3. Quantum algorithms

3.1. Quantum-accelerated linear solvers

A very popular classical algorithms for solving linear systems of the form (12) with symmetric positive-definite s -sparse system matrix A_h is the conjugate gradient (CG) method [24]. Its time complexity is $\mathcal{O}(Ns \kappa \log(1/\epsilon))$ [25], where ϵ is the desired accuracy of the solution in some norm, i.e. $\|A_h y_h - b_h\| < \epsilon$.

The first quantum algorithm for solving linear systems of equations was developed by Harrow et al. in [26] with time complexity $\mathcal{O}(\log(N)s^2 \kappa^2 / \epsilon)$. Improved variants have been proposed by Ambainis in [27] ($\mathcal{O}(\log(N)s^2 \kappa / \epsilon)$) and Childs et al. in [28] ($\mathcal{O}(s \kappa \text{polylog}(s \kappa / \epsilon))$). The exponential speed up of the Quantum Linear System Algorithm (QLSA) comes at the price that it does not return vector y_h , but the scalar quantity $\langle y_h | M_h | y_h \rangle$, where

M_h is some sparse matrix operator. This nifty detail makes the QLSA quite impractical as a general-purpose acceleration technique for solving linear systems of equations but particularly attractive for use in the ADO framework outlined in Section 2.1.

Without going into the technical details, we remark that the matrices A_h and M_h resulting from a finite element discretization of the academic model problem (7)–(8) satisfy all prerequisites of the QLSA, which can therefore be used as quantum accelerator for solving (12) and returning (14) as output.

Remarks. Many problems of practical interest do not yield symmetric system matrices if the spatial discretization is applied directly. An alternative to solving the augmented problem that is symmetrized algebraically, i.e.

$$\begin{pmatrix} 0 & A_h \\ A_h^\dagger & 0 \end{pmatrix} \begin{pmatrix} x_h \\ y_h \end{pmatrix} = \begin{pmatrix} b_h \\ 0 \end{pmatrix} \quad (15)$$

is to adopt special discretization schemes like the Least-Squares/Galerkin finite element method [29] that symmetrize the problem at the continuous level.

Recently, Wossnig et al. [30] extended the QLSA to dense matrices with time complexity $\mathcal{O}(\sqrt{N}\text{polylog}(N)\kappa^2/\epsilon)$. An alternative solution approach for solving BVPs numerically is the boundary element method (BEM) [31], which yields dense system matrices by design and is widely used for the simulation of acoustics problems. The most efficient classical solution approach for solving the BEM is the fast multipole method, which has runtime $\mathcal{O}(N)$ in the best case, that is, for non-oscillatory kernels, and $\mathcal{O}(N\log(N))$ otherwise and is therefore less efficient than its quantum-accelerated counterpart.

Conclusion. The above discussion suggests that the QLSA has great potential to deliver exponential speed up over classical approaches for solving certain types of BVPs if the quantity of interest is not the solution vector but a linear cost functional of the form (14). This is the case for many ADO problems as outlined in Section 2.1 but also in the context of goal-oriented error estimation.

The interested reader is referred to the QLSA primer by Devovic et al. [25] for a description of the different variants of the algorithm and to the research article by Cao et al. [32] for a possible quantum circuit design for the QLSA.

Let us finally mention two interesting publications on using quantum algorithms for accelerating the solution of the Poisson problem discretized by the FEM. While Clader et al. [33] report exponential speed up, Montanaro and Pallister [34] observe that the exponential quantum advantage can sometimes disappear. They see the quantum advantage only for problems, where the solution has larger higher order derivatives or if the spatial dimension is high.

Last but not least we remark that many applications of practical interest involve nonlinear PDEs like the Navier-Stokes equations and/or cost functionals like the lift-to-drag coefficient (4) that cannot be cast into the form (14) directly. In both cases, the problem at hand needs to be approximated by a suitable linearization technique, which requires further investigation.

3.2. Quantum-accelerated optimization

Let the current set of design parameters α^k in (5)–(6) be in the neighborhood of a (possibly locally) optimal solution α^* so that the cost functional $\mathcal{J}(\alpha)$ can be approximated by the multi-variate Taylor series expansion

$$\begin{aligned} \mathcal{J}(\alpha^k) - \mathcal{J}(\alpha^*) &= \frac{1}{2} \sum_{i,j=1}^M \mathcal{H}_{ij}(\alpha_i^k - \alpha_i^*)(\alpha_j^k - \alpha_j^*) \\ &+ \mathcal{O}(\|\alpha^k - \alpha^*\|^3) \end{aligned} \quad (16)$$

Here, $\mathcal{H}_{ij} = \partial^2 \mathcal{J} / \partial \alpha_i \partial \alpha_j |_{\alpha=\alpha^*}$ denotes the matrix of second derivatives of the cost functional taken with respect to the design variables and evaluated at the equilibrium point α^* . The matrix of first derivatives vanishes since $\nabla \mathcal{J}|_{\alpha=\alpha^*} \equiv 0$. By neglecting all higher-order terms, the original minimization problem (5) can be solved by finding the minimum to the positive-definite quadratic form

$$Q(\alpha) := \frac{1}{2} \sum_{i,j=1}^M \mathcal{H}_{ij}(\alpha_i^k - \alpha_i^*)(\alpha_j^k - \alpha_j^*) \quad (17)$$

Classical algorithms need at least $\mathcal{O}(M^2)$ queries [35] to achieve this goal, where M is the number of design variables. However, quantum computing makes it possible to determine the minimum of a quadratic form in only $\mathcal{O}(M)$ quantum queries [36]. The key ingredient to this significant reduction in query complexity is an efficient quantum algorithm for estimating gradients [37,38].

As stated above, a common shortcoming of gradient-based methods for solving continuous global optimization problems is the risk to get trapped in a local minimum solution from which there is no escape. A recent development in classical ADO technologies is to combine gradient-based methods with gradient-free ones into hybrid approaches [15,16]. Interestingly enough, the idea of hybrid methods has its counterpart in quantum-based optimization [39]. The central idea is to determine locally optimal solutions by using (classical) methods and escape from non-global solutions making use of Grover's search algorithm.

Conclusion. There is evidence that also the outer optimization loop of the overall ADO problem can be accelerated by using quantum optimization algorithms.

4. Conclusions and outlook

Based on an inventory of existing quantum algorithms for solving linear systems of equations and continuous optimization problems, we have sketched a conceptual framework for accelerating the solution of automated design optimization problems with the aid of quantum computers. Admittedly, the suggested approach leaves many open questions and unresolved technical issues that need to be addressed in future research activities. An important aspect is the efficient encoding of real-valued data that is less wasteful on (qu)bits than the IEEE-754 floating-point standard, which requires 32 and 64 (qu)bits to encode a single floating-point number in single and double precision, respectively.

The recently introduced Universal NUMber format [40] adopts a projection of the real numbers onto the unit circle and represents selected discrete angles with specific bit patterns. UNUM claims to achieve the same accuracy as IEEE-754 with less bits. It seems plausible that this concept can be carried over to the Bloch sphere, thereby using quantum rotations to encode real-numbers in a *single* qubit. It is, of course, not possible to measure the so-encoded quantities but they might still be helpful for storing intermediate values, especially, if the overall problem is formulated in terms of 'Is α_1 better than α_2 '?

Another unresolved issue are the non-trivial preparatory steps of the QLSA and quantum-accelerated optimization, which complicate their direct interplay.

In summary, we consider this paper a first step to make the scientific computing community aware of the possibilities of quantum-accelerated computing and to stimulate interdisciplinary research in practical quantum computing.

Acknowledgments

The authors would like to thank the reviewers for their valuable feedback as well as Koen Bertels and Carmina Garcia Almudever for many fruitful discussions.

References

- [1] L. Dagum, R. Menon, OpenMP: an industry standard API for shared-memory programming, *Comput. Sci. Eng., IEEE* 5 (1) (1998) 46–55.
- [2] L. Clarke, I. Glendinning, R. Hempel, The MPI message passing interface standard, in: K. Decker, R. Rehmman (Eds.), *Programming Environments for Massively Parallel Distributed Systems*, Monte Verità (Proceedings of the Centro Stefano Franscini Ascona), Birkhäuser, Basel, 1994, pp. 213–218.
- [3] M. Feldman, Google offers glimpse of third-generation TPU processor, 2018. (Top 500 (Retrieved 5 January 2019): <https://www.top500.org/news/google-offers-glimpse-of-third-generation-tpu-processor/>).
- [4] M. Möller, C. Vuik, On the impact of quantum computing technology on future developments in high-performance scientific computing, *Ethic. Inf. Technol.* 19 (4) (2017) 253–269, doi:10.1007/s10676-017-9438-0.
- [5] J. Preskill, Quantum computing and the entanglement frontier, 2012 *arXiv:1203.5813*.
- [6] W. Knight, IBM raises the bar with a 50-Qubit quantum computer, 2017. (MIT Technology Review (Retrieved 5 January 2019): <https://www.technologyreview.com/s/609451/ibm-raises-the-bar-with-a-50-qubit-quantum-computer/>).
- [7] J. Kelly, A preview of bristlecone, Googles new quantum processor, 2018. (Google AI Blog (Retrieved 5 January 2019): <https://ai.googleblog.com/2018/03/a-preview-of-bristlecone-googles-new.html>).
- [8] C. Rigetti, The Rigetti 128-qubit chip and what it means for quantum, 2018. (Rigetti Computing (Retrieved 5 January 2019): <https://medium.com/rigetti/the-rigetti-128-qubit-chip-and-what-it-means-for-quantum-df757d1b71ea>).
- [9] A. Quarteroni, A. Valli, *Domain Decomposition Methods for Partial Differential Equations*, Oxford Science Publications, Oxford, 1999.
- [10] J.A. Park, The concept of transition in quantum mechanics, *Found. Phys.* 1 (1) (1970) 23–33.
- [11] C.H. Bennett, G. Brassard, C. Crépeau, R. Jozsa, A. Peres, W.K. Wootters, Teleporting an unknown quantum state via dual classical and Einstein-Podolsky-Rosen channels, *Phys. Rev. Lett.* 70 (13) (1993) 1895, doi:10.1103/PhysRevLett.70.1895.
- [12] L. Gyongyosi, S. Imre, H.V. Nguyen, A survey on quantum channel capacities, *IEEE Commun. Surv. Tutor.* 20 (2) (2018) 1149–1205.
- [13] S. Sivanandam, S. Deepa, *Introduction to Genetic Algorithms*, Springer-Verlag, Berlin Heidelberg, 2008, doi:10.1007/978-3-540-73190-0.
- [14] J. Nocedal, S. Wright, *Numerical Optimization*, Springer Series in Operations Research and Financial Engineering, 2, Springer-Verlag New York, 2006, doi:10.1007/978-0-387-40065-5.
- [15] V. Kelner, F. Capitanescu, O. Léonard, L. Wehenkel, A hybrid optimization technique coupling an evolutionary and a local search algorithm, *J. Comput. Appl. Math.* 215 (2) (2008) 448–456, doi:10.1016/j.cam.2006.03.048.
- [16] T. Verstraete, *Priv. Commun.*, 2017.
- [17] R.J. LeVeque, *Finite Difference Methods for Ordinary and Partial Differential Equations*, Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, 2007.
- [18] S. Patankar, *Numerical Heat Transfer and Fluid Flow*, Series in Computational and Physical Processes in Mechanics and Thermal Sciences, Hemisphere Publishing Corporation, 1980.
- [19] O. Zienkiewicz, R. Taylor, *The Finite Element Method*, 5th, Butterworth-Heinemann, Oxford, 2000.
- [20] Y. Saad, *Iterative Methods for Sparse Linear Systems* 2nd edition, Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, 2003.
- [21] F. Vermolen, A. Segal, On an integration rule for products of barycentric coordinates over simplexes in \mathbb{R}^n , *J. Comput. Appl. Math.* 330 (2018) 289–294, doi:10.1016/j.cam.2017.09.013.
- [22] G. Strang, G. Fix, *An Analysis of the Finite Element Method*, Prentice-Hall, Inc., Englewood Cliffs, 1973.
- [23] Z.-C. Li, H.-T. Huang, Effective condition number for the finite element method using local mesh refinements, *Appl. Numer. Math.* 59 (8) (2009) 1779–1795, doi:10.1016/j.apnum.2009.01.006.
- [24] J.R. Shewchuk, *An Introduction to the Conjugate Gradient Method Without the Agonizing Pain*, Technical Report, Carnegie Mellon University, Pittsburgh, PA, USA, 1994.
- [25] D. Dervovic, M. Herbster, P. Mountney, S. Severini, N. Usher, L. Wossnig, Quantum linear systems algorithms: a primer, 2018 *arXiv:1802.08227*.
- [26] A.W. Harrow, A. Hassidim, S. Lloyd, Quantum algorithm for linear systems of equations, *Phys. Rev. Lett.* 103 (2009) 150502, doi:10.1103/PhysRevLett.103.150502.
- [27] A. Ambainis, Variable time amplitude amplification and a faster quantum algorithm for solving systems of linear equations, in: D. Christoph, W. Thomas (Eds.), *Proceedings of the 29th Symposium on Theoretical Aspects of Computer Science (STACS'12)*, 2012, pp. 636–647.
- [28] A.M. Childs, R. Kothari, R.D. Somma, Quantum algorithm for systems of linear equations with exponentially improved dependence on precision, *SIAM J. Comput.* 46 (6) (2017) 1920–1950, doi:10.1137/16M1087072.
- [29] P.V. Bochev, M.D. Gunzburger, Finite element methods of least-squares type, *SIAM Rev.* 40 (4) (1998) 789–837, doi:10.1137/S0036144597321156.
- [30] L. Wossnig, Z. Zhao, A. Prakash, A quantum linear system algorithm for dense matrices, *Phys. Rev. Lett.* 120 (5) (2018) 050502, doi:10.1103/PhysRevLett.120.050502.
- [31] S. Sauter, C. Schwab, *Boundary Element Methods*, Springer Series in Computational Mathematics, 39, Springer-Verlag Berlin Heidelberg, 2011, doi:10.1007/978-3-540-68093-2.
- [32] Y. Cao, A. Daskin, S. Frankel, S. Kais, Quantum circuit design for solving linear systems of equations, *Mol. Phys.* 110 (15–16) (2012) 1675–1680, doi:10.1080/00268976.2012.668289.
- [33] B.D. Clader, B.C. Jacobs, C.R. Sprouse, Preconditioned quantum linear system algorithm, *Phys. Rev. Lett.* 110 (2013) 250504, doi:10.1103/PhysRevLett.110.250504.
- [34] A. Montanaro, S. Pallister, Quantum algorithms and the finite element method, *Phys. Rev. A* 93 (2016) 032324, doi:10.1103/PhysRevA.93.032324.
- [35] A. Yao, *On computing the minima of quadratic forms*, *ACM Symp. Theor. Comput.* (1975) 23–26.
- [36] S.P. Jordan, *Quantum Computation Beyond the Circuit Model*, Massachusetts Institute of Technology, Department of Physics, 2008 Ph.D. thesis.
- [37] S. Jordan, Fast quantum algorithm for numerical gradient estimation, *Phys. Rev. Lett.* 95 (2005) 050501, doi:10.1103/PhysRevLett.95.050501.
- [38] A. Gilyén, S. Arunachalam, N. Wiebe, Optimizing quantum optimization algorithms via faster quantum gradient computation, 2017 *arXiv:1711.00465*.
- [39] P. Lara, R. Portugal, C. Lavor, A new hybrid classical-quantum algorithm for continuous global optimization problems, *J. Global Optim.* 60 (2) (2014) 317–331, doi:10.1007/s10898-013-0112-8.
- [40] J.L. Gustafson, *The End of Error: Unum Computing*, Computational Science, Chapman & Hall / CRC, 2015.



Dr. rer. nat. Matthias Möller is Assistant Professor in the Numerical Analysis Group of the Department of Applied Mathematics at Delft University of Technology. He obtained a Diploma in Mathematics from the Faculty of Mathematics at TU Dortmund University, Germany, in 2003 and received a PhD from the same institution in 2008. His research interests focus in numerical methods for partial differential equations and their efficient implementation on heterogeneous high-performance computing platforms. In particular, he has been working on high-resolution adaptive finite element methods and fast iterative solution techniques for convection-dominated transport problems and compressible flows. He is more-over active in the field of isogeometric analysis, where he develops high-order isogeometric solvers for compressible flows with special focus on exploiting accelerator-technologies like GPUs and FPGAs to speed-up computations. Recently, he has extended his research interests to quantum computing looking at this emerging technology from the perspective of an end-user, who is interested in adopting quantum computers as accelerator devices in a heterogeneous high-performance compute platform.



Prof. ir. Cornelis Vuik is Full Professor of Numerical Analysis at the Delft University of Technology. He is Director of TU Delft Institute for Computational Science and Engineering (DCSE), which is represented within six different faculties from TU Delft with about forty research groups and more than three hundred faculty members being connected to, and actively involved in DCSE and its activities. Cornelis Vuik is moreover the Scientific Director of 4TU.AMI Applied Mathematics Institute. This is the Applied Mathematics Institute of the Dutch 4TU federation. There is a close connection between 4TU.AMI and Research Center Matheon in Berlin. Together they form the largest group of applied mathematicians in Europe. Prof. Vuik is an expert on numerical methods for partial differential equations, fast and robust linear and nonlinear solvers and high-performance computing. He has published more than 160 ISI papers in international journals and has an ISI H-index of 24. Within Google scholar a total of 5950 citations are found and his H-index is 38.