

Computing Competency for Civil Engineering Graduates: Recent Updates and Developments in Saudi Arabia and the US

Danish Ahmed¹, Mohammed Nayeemuddin¹, Tahar Ayadat¹ & Andi Asiz¹

¹ Prince Mohammad Bin Fahd University, Al Khobar, Saudi Arabia

Correspondence: Tahar Ayadat, Prince Mohammad Bin Fahd University, Al Khobar, Saudi Arabia.

Received: April 19, 2021

Accepted: June 9, 2021

Online Published: June 11, 2021

doi:10.5430/ijhe.v10n6p57

URL: <https://doi.org/10.5430/ijhe.v10n6p57>

Abstract

This paper discusses recent updates and developments of computing-based courses in the civil engineering discipline. Competency in computing is one of the most important capabilities for university graduates to obtain given the rapid development of computer technology in professional work. Civil engineering is no exception. In fact, many contemporary civil engineering projects require a high degree of computing skills, ranging from performing basic office work to programming for decision support system application in controlling flood water gates to executing construction automation via digital printing technology. However, the curriculum content for computing in civil engineering has been developmentally stagnant in the past several decades. This could be partly due to learning outcomes for civil engineering graduates, which do not explicitly mention a certain degree of achievement with respect to computing skills. Several computing-based courses offered in various civil engineering programs across Saudi Arabia and the US were examined, and their contents were compared to recent survey results administered by the American Society of Civil Engineering Technical Committee on Computing and Information Technology. The discussion is extended by examining technical courses offered in the Civil Engineering Program in Prince Mohammad Bin Fahd University with respect to computing skills. The outcomes of this study are expected to give input and suggestions for future upgrades of computing-based courses offered within the civil engineering curriculum.

Keywords: computing skills, programming, engineering software, curriculum, civil engineering

1. Introduction

Computer literacy or computing skill is one of the most important capabilities every university graduate must possess. Nearly every professional occupation requires computers to run its business, from creating professional reports using word processors to operating machinery equipment using computer-controlled numeric programs. Engineering professions, such as civil engineering, have developed and advanced their business through the problem-solving capability of the various simple and complex mathematical equations they routinely apply. When it comes to math and science problem solving, computers, even in smaller forms, such as calculators, greatly assist in the usefulness and meaning fullness of mathematical formulae. Unfortunately, other than computer or information technology majors, most university curricula continue to devote minimal learning hours for developing computing skills. The civil engineering discipline is no exception in this regard. Historically, computing in civil engineering emerged immediately following the invention of computers, as it substantially assisted in solving complex mathematical calculations. One of the pioneers in civil engineering computing was Professor Clough of the Massachusetts Institute of Technology, who developed a computer algorithm in the early 1960s for solving large algebraic simultaneous equations used in structural mechanics (Clough, 1958). Professor Wilson of the University of California at Berkeley, who developed a computer program using the FORTRAN compiler for conducting two-dimensional structural analyses of multi-story buildings, complemented his work (Wilson, 1963). Professor Wilson is well known in the structural engineering profession as the founding father of computer-based finite element analysis, which is still widely used in modern integrated computer-aided design across many engineering disciplines. Since then, a few finite element software programs have been developed to perform daily structural design and analysis of buildings under various loads, particularly complex earthquake-dynamic loads. The finite element method is a powerful numerical tool used to solve partial differential equations encountered in many natural science and mechanics phenomena and is moreover the basis for advancing computing technology in many engineering disciplines. Due to

the high importance of computing for civil engineering practitioners and researchers, the American Society of Civil Engineering allocated one of its journal publications to computer-related research, the *Journal of Computing in Civil Engineering*, which has been publishing research since the late 1980s (Abudayyeh et al., 2006).

Contemporary civil engineers use computers to solve daily work ranging from design and analysis to construction execution to the operation and monitoring of various infrastructure projects. Intelligence-based computing in civil engineering, such as the adaptive method in computation, expert systems with neural networks or genetic algorithms, integrated GIS-based construction, and construction automation, is one of the research project areas that has grown rapidly as computers have become more powerful (Asif et al., 2015; Ehsan, 2019; Ibraheem et al., 2012; Kheyfets & Vasilieva, 2017; Pengzhen et al., 2012; Sampaio et al., 2010; Vinay Chandwani et al., 2013; Zhou, 2016). To cope with the rapid development in computer technology, civil engineering graduates must possess adequate competency in computing. However, the curriculum content for teaching computing-based courses has remained stagnant in the past several decades, relying heavily on traditional programming-based teaching as main criteria for computing literacy (Grigg et al., 2004; Grigg et al., 2005; Vergara et al., 2015). This could be partly due to learning outcomes for civil engineering programs that do not explicitly mention the certain degree of achievement needed for computing skills. For example, the Accreditation Board for Engineering and Technology (ABET), one of the mainstream accreditors in US engineering schools, does not require engineering programs to directly assess computing skill criteria. The ABET does, however, give flexibility to attach computing skills to one or more student learning outcomes, such as design ability or the capacity to acquire advanced learning skills, i.e., developing life-long learning strategies (Ayadat et al., 2020). As few studies have addressed the development of computing competency for civil engineering graduates, the current research was intended to serve as an important reminder for civil engineering educators to prepare graduates to meet computing competency criteria by revisiting and updating the current curriculum, including its computing learning outcomes. As with other engineering programs, the civil engineering program typically offers three levels of computing-based skills. The first level typically concerns the ability to use basic computer technology, such as word processing, spreadsheets, and electronic communication. The second level involves the introduction to programming using low- or high-level computer languages. The third (advanced) level concerns the utilization of computer software in engineering design. The first and third levels can be embedded within several general or civil engineering courses, and these include running software for engineering design as well as developing limited coding for solving advanced mathematical equations used in engineering.

2. Literature Review

Relative to research activities in civil engineering computing, which are abundant, literature in the area of computing education in the civil engineering curriculum is scarce. This is partly due to the limited focus on computer literacy in the curriculum despite the fact that nearly every course needs computer software to either learn the subject or facilitate the learning process. In the early stage of computer development, engineering professors correlated computer literacy with programming skills due to the very limited software packages and programs available during the period. In addition to basic programming skills, Rasdorf (1985) recommended inserting more computer science subjects into the curriculum and emphasized the implementation of more programming exercises in civil engineering applications aside from solving complex mathematical equations. Rasdorf suggested focusing on basic programming skills, including data structure, program control, and program organization, so that students would not be too concerned with computer science-oriented learning outcomes but would also retain knowledge on how to develop basic programming skill applications in civil engineering areas. At that time, FORTRAN (Formula Translator) was the most common computer language used by civil engineering students in their programming courses due to its straightforward application in solving mathematical formulae. Ultimately, Rasdorf recommended that computer science applications be cautiously justified against basic principles of civil engineering that every student and graduate had already mastered. As computer software packages became more available for practical use, Gifford (1987) suggested the ethical application of computers in civil engineering disciplines to avoid overconfidence in computer outcomes without proper justification. Engineers at that time called “black-box syndrome,” a situation in which there is a heavy reliance on computational skills instead of on principles underlying the development of particular computer software, resulting in black box “data in – solution out.” To prevent the black-box syndrome, Gifford recommended that civil engineering students must understand the basic philosophy of the software and its limitations, including its operational assumptions and its suitability in certain applications. Furthermore, he suggested that a high degree of professional and ethical behaviors on the part of software users equipped with extensive knowledge and training can prevent the misuse of computers in civil engineering.

In the late 1980s and early 1990s, when engineering software development had advanced with graphical interfaces, several educators in the US conducted surveys to review and update the status of computing skills development

within the civil engineering curriculum. Baker and Rix (1992) reviewed the relevance of computing-based courses via surveys distributed to undergraduate students in US civil engineering programs and to alumni specific to one university, the Georgia Institute of Technology in Atlanta. The intention of the survey was to gauge the students' satisfaction with the computing skills learned in the curriculum, including the adequacy of the associated hardware and software, and to determine the degree to which alumni were satisfied with the computing skills gained during their undergraduate study in terms of their realization in the professional workplace. The study results suggested that due to diverse levels of computer literacy among students upon their arrival to university from high school, computer courses needed to be provided at an early stage, i.e., to offer a course that can reach a common denominator of computing skills. The survey results obtained from the alumni indicated that civil engineering was behind other engineering disciplines in terms of utilizing computer software in professional work settings, and as such a mutual collaborative approach between industry and university should be developed to address the adequacy issue of computer hardware and software. Another recommendation from this study was to offer computing skills in three levels: (1) learning the basic technology of using computer software (e.g., spreadsheets, word processors), (2) learning computer literacy awareness (e.g., ethics, terminology, computing techniques), and (3) learning computer programming. Although according to the alumni survey responses, civil engineers do not need extensive programming skills in practice, such skills can nonetheless increase engineers' awareness of how to develop software used in their daily engineering work in principle.

A recent survey distributed to educators about computing in various civil engineering programs across the US indicated that operating basic office computers (e.g., spreadsheets, word processing, and presentation packages) was the most important skill set for computing followed by utilizing specialized engineering software (Gerber et al., 2004). Programming skills were considered less important (ranked 8 out of 10) for student competency in computing. The survey distributed to civil engineering practitioners showed similar results, ranking basic office computer skills in the top three in terms of importance followed by computer-aided drafting. The practitioners also ranked programming skills close to the bottom (8 out of 10). Despite such a low ranking in the survey, all civil engineering programs have retained programming-related courses in their curricula, as shown in Table 1 (Part A). Educators and practitioners ranked the coverage of computing skills, in descending order, within civil engineering curricula as follows: computer-aided drafting, utilization of specialized engineering software, programming, and spreadsheet use (Gerber et al., 2004). Basic office computer skills, such as word processing and presentation packages, were not on the top of the list due to some degree of familiarity with those skills among students. Students normally obtain these skills from low-level non-engineering courses, such as writing and presentation in English language courses.

Table 1(Part A). (Universities in Saudi Arabia): Computing-based courses in civil engineering programs.

No	University and program name	Credit hours for the degree	Computing course name (credit hours)	Course prerequisites	Course description
1	KFUPM (King Fahd University of Petroleum and Minerals) – Civil Engineering	160	1. ICS 103 Computer Programming in C (3) 2. CE 318 Numerical and Statistical Methods in Civil Engineering (3)	None ICS 103 Computer Programming in C, MATH 208 Introduction to Differential Equations and Linear Algebra	Computer hardware, software, programming in C, programming technique, problem solving, and algorithm development Introduction to numerical methods; error analysis, linear and nonlinear equations, numerical integration, numerical solutions of ordinary differential equations; curve fitting, interpolation, statistical methods, descriptive statistics, probability distributions, variance and regression; introduction to linear programming and development of computer programs for civil engineering practices
2	KSU (King Saud University) – Civil	165	1. GE 209 Computer Programming (3)	None	Computer programming for solving engineering problems in MATLAB environment by utilizing all kinds

Engineering			of functions		
3	Qassim University –Civil Engineering	139	2. MATH 254 Numerical Methods (3)	MATH 107 Vectors and Matrices	Numerical methods for nonlinear equations; direct and iterative methods for linear equations with error estimation; polynomial interpolation; numerical differentiation and integration; numerical solution of ordinary differential equations
			1. CSC 209 Computer Programming (3)	None	Introduction to computers and computing fundamentals in JAVA, and learning all kinds of functions for engineering applications
			2. CE 317 Computer Applications (3)	CE 353 Geotechnical Engineering, CE 315 Reinforced Concrete	Problem formulation; preparing problem models; constitutive modeling; using FEM-based software packages in design and solving engineering problems—for example, SAP2000, PLAXIS, Geoslope Suite, ANSYS, STAD Pro, Mud Flow, Pipe Net, etc.
4	KAA (King Abdulaziz University) –Civil Engineering	155	1. CPIT 110 Problem-Solving and Programming (3)	None	The main objective of this course is to teach students the basics of constructing algorithms and programming languages by utilizing all of their functions.
			2. EE 201 Structured Computer Programming (2)	MATH 110 General Mathematic, CPIT 110 Problem-Solving and Programming	Describes the basics of MATLAB, applies MATLAB to solve engineering problems, describes the fundamentals of programming, designs simple programs, and modularizes the programs using functions
5	Imam Abdulrahman Bin Faisal University –Construction Engineering	136	1. COMP 131 Computer Skills (2)	None	The focus of this course is to provide main concepts and terminologies of information technology in practical applications and also to teach word processing, spreadsheets, and presentations
			2. COMP 212 Computer Programming (2)	None	This course provides concepts of algorithms, programming language, and problem-solving skills
			3. MATH 472 Numerical Methods (3)	MATH 331 Differential Equations	Fixed-point, Bisection, Fasle, Newton-Raphson, and Secant methods; linear and non-linear equations using iterative methods of Jacobi, Gauss-Seidel, SOR, Gaussian elimination technique, LU- factorization, numerical integration; first-order differential equations using Taylor’s, Runge-Kutta, Euler’s, and Adams Bashforth methods; and parabolic,

				hyperbolic, and elliptic partial differential equations
		1. CS 100 Computer Principles (3)	None	Computer skills, introduction to programming
		2. EE 201Structured Computer Programming (2)	MATH 110 General Mathematics I, CS 100 Computer Principles	The basics of MATLAB and its built-in functions for the computation of mathematical formulae. The user-defined functions for solving engineering problem cases and the development of various computer-structured programs via the m-files of MATLAB for engineering practice
6	Northern Border University –Civil Engineering	162	3. EE 332 Numerical Methods in Engineering (3)	MATH 204 Differential Equations I, EE 201 Structured Computer Programming
			4. CE 343 Computer Applications in Civil Engineering (3)	CE 342 Reinforced Concrete Design I, CE 203 Civil Drawing
			1. Introduction to Computer Programming (2)	None
7	Taif University –Civil Engineering	153	2. 2023206-3 Numerical Analysis (3)	2023103-4 Differential Equations
			1. CSC 001 Computer Skills and Its Application (3)	None
8	University of Tabuk – Civil Engineering	166	2. CE 494 Computer Application for Civil Engineering (3)	ENG 206 Engineering Mechanics, MATH 241 Linear Algebra
9	Majmaah University – Civil and Environmental Engineering	136	1. CEN 209 Computer Programming for Civil Engineering (3)	None

10	Jubail University College – Civil Engineering	131	2. MATH 254 Numerical Methods (3)	MATH 204 Differential Equations	Resolution of non-linear equations by numerical methods, estimation of the errors committed by these methods and rates of convergence of iterative methods. Resolution of linear equations by direct methods and recurring methods, and calculation of the errors related to these methods. Interpolation by using the polynomials and the formula of errors committed by this interpolation. Numerical differentiation and integration, including related errors. Introduction to numerical solutions of ordinary differential equations
			3. CE 425 Computer Applications in Structural Engineering (2)	CEN 209 Computer Programming for Civil Engineering	Writing computer programs using Basic or Visual Basic, C++ or Excel in civil engineering (structural analysis, reinforced concrete design, steel structures design, foundation engineering, hydraulics and water engineering), and training on using software (such as STAAD-III, Excel, AutoCAD). Also, utilization of internet resources in civil engineering
			1. CS 125 Computer Programming (3)	MATH 111 Calculus I	Introduction to engineering problem-solving techniques using C++ programming language; employment of a problem-solving methodology to consider a diverse range of challenging engineering issues. The emphasis on engineering and scientific problem solving remains an essential constituent of the course. It covers software engineering models, focusing on the design and implementation of user-friendly and reusable computer solutions and structured programming
			2. MATH 314 Numerical Methods (3)	CS 125 Computer Programming, MATH 211 Calculus III	Roots of nonlinear equations; solution of systems of linear and nonlinear algebraic equations; numerical differentiation and integration; interpolation, extrapolation, and approximation; least-squares approximation and regression analysis; numerical solution of ordinary differential equations; introduction to error analysis; engineering case studies

3. Methodology

The main method used in this study was the comparative analysis of recent computing courses offered in various civil engineering programs in Saudi Arabia and the US. The course syllabi of these programs, including descriptions, prerequisites, and credit hours, were listed and compared to those of the recent survey results administered by the American Society of Civil Engineering (ASCE) Technical Committee on Computing and Information Technology (Gerber et al., 2004; Osama et al., 2004). All programs described in this study are accredited by the ABET Engineering Accreditation Commission. The Saudi universities discussed in this study are top-10 public universities that offer civil engineering programs. Due to the large number of civil engineering programs offered in the US, the authors included 10 universities within the top tier (ranked 1–50 according to US World Ranking). The authors extended the discussion by examining technical courses offered in the Civil Engineering Program at Prince Mohammad Bin Fahd University (PMU) with respect to computing skills. The authors also described key computing skills embedded in the courses, including the utilization of modern computer software for engineering design. Major outcomes of this study included input and suggestions for future upgrades of computing-based courses offered within the civil engineering curriculum.

4. Computing Courses in Civil Engineering

Table 1 shows a list of civil engineering programs in Saudi Arabia and the US that offer computing skills along with the course name, course description, and course prerequisites. More than 20 universities in Saudi Arabia offer civil and environmental engineering programs, while a thousand universities in the US offer civil and environmental engineering programs. The 10 Saudi universities were selected based on their popularity, whereas the 10 US universities were chosen based on representation of high, medium, and low tier of research-based universities. All of the Saudi Arabian universities listed in the table are both popular and large public universities that offer a civil engineering program, including environmental and construction engineering programs. These universities have secured national and international accreditation from the National Commission on Academic Assessment and Accreditation (NCAAA) and the ABET, respectively. Programming and computation with computers are the only courses listed here, and not the other computing courses (e.g., word processing, software utilization) because these course are arguably one of the main criteria for recognizing, applying, and advancing computing skills. The total credit hours required to confer the BS degree are also provided to observe the percentage of computing skill hours allocated within the curriculum.

The authors observed that all civil engineering programs offered basic skills in computer programming using applicable programming languages, and that the majority of the programs allocated one or two additional courses in computer programming with more applications in mathematical problem solving in engineering. In general, the total credit hours spent by students for developing programming skills was less than 5% of the total credit hours for obtaining the BS degree, i.e., between 3 to 6 credit hours. Most of the programs use low-level computer language C or C++ to train students in basic coding in programming. This is partly due to the fact that most of the instructors teaching programming courses are from computer science programs and have a high degree of familiarity with this language. The main challenge for civil engineering students taking the course along with computer science students is the depth of programming skills, which should be different since civil engineering students must focus on applying their engineering principles in problem solving via computer use (Bowen, 2004; List, 2007; Sun, 2011).

Of the 10 Saudi universities listed in Table 1 (Part A), three offer explicit basic computing skills, such as learning the concepts and terminologies used in computer and information technology, operating word processors, using spreadsheets, and making presentations. In contrast, in the US, none of the universities offer explicit basic skills in computing. This could be due to the uniform expectations of computing skills developed in US high school graduates relative to those of Saudi graduates. It could also be due to the more advanced computer technology available in US society compared to that available in Saudi society. The problem of incorporating basic computing skills as a standalone course is limited space in the curriculum and the attachment of introductory courses at the freshmen level to the program, such as English (language) communication courses, introduction to engineering course, and math (calculus) courses.

Courses such as numerical methods or computational methods are taught using high-level computer languages equipped with equation solvers, such as MATLAB, MATCAD, Maple, and Mathematica. In this type of course, programming skills are taught after students are able to solve close-formed mathematical equations analytically by “hand/manual calculation,” and the knowledge obtained from the previous basic programming course is then used to develop algorithms for solving equations numerically. At this level, students are expected to learn to perform comparative analysis between “manual calculation” results versus numerical computer output. Most of the course

prerequisites from introductory computing courses are math/calculus courses, and upper-level computing courses required student to pass the introductory computing.

5. Discussion

At PMU, computing literacy is one of the performance indicators for student graduates in achieving technological competence, including using modern computer technologies to acquire information, communicate, solve problems, and produce intended results (Petuhova, 2015). Alongside official academic transcripts, a rating of technological competency (e.g., outstanding, excellent, good, and novice) is given for individual students by quantifying a series of course learning outcome assessments that incorporate basic and advanced computing skills. While at the Civil Engineering Program level, none of the ABET student learning outcomes directly address competency in computing. As in the university competency level, specific performance indicators related to computing skills are embedded into some of the student learning outcomes. For example, the ability of students to run computer software in design can be categorized under either “ability to perform engineering design” or “ability to develop life-long learning strategy” outcomes (Petuhova, 2015). The ABET-accredited Civil Engineering Program at PMU offers a bachelor’s degree with 139 credit hours (50 courses) that can be completed within four years (8 semesters, including one summer internship). Table 2 presents technical courses offered under the PMU Civil Engineering Program along with the acquired computing skills. It should be noted that these computing skills are taught and assessed through the learning outcomes of the respective courses. It can be seen that applying computer software in civil engineering occupies predominates in the development of computing skills, covering more than 74% of the technical courses offered. Not all technical courses include computing skills in their learning outcomes (e.g., Engineering Mechanics, Thermodynamics, Introduction to Fluid Mechanics, Engineering Economy). Programming skills are used explicitly in the two-credit-hour Engineering Computing course, representing 1.4% of the total credit hours.

In addition to utilizing statistical software packages, a limited coding exercise using a spreadsheet is applied in the Engineering Probability and Statistics course to obtain and analyze statistical parameters. Small-scale application of coding is used in the Hydraulic Engineering course for solving high-degree polynomials encountered in fluid flow. Geoslope is used in the Introduction to Geotechnical Engineering course for analyzing slope stability. Furthermore, stability analysis and design problems in the Foundation Analysis and Design course are addressed using spreadsheets developed for this purpose. The Structural Analysis course requires students to solve simultaneous (algebraic) equations either by using the MATLAB solver directly or by developing a simple coding for it. MATLAB has become a common software program for civil engineering applications due to its flexibility in providing solutions via direct solvers or creating simple and complex coding (Dai, 2006; Tavakoli & Klika, 1991). Basic computing skills for performing office-related work (e.g., report writing, presentation) are taught in nearly 35% of the technical courses. The lower percentage of basic computing skills relative to that of advance skills is attributable to the fact that most students already possess some degree of computing literacy from their previous education (e.g., high school or computer training) or from other non-technical courses offered by non-engineering programs within the civil engineering curriculum.

The GEEN 2211 – Engineering Computing course is offered as the first-level computer-based programming course for PMU engineering students. It is a two-credit-hour course, including laboratory hours, and is delivered at many sections across engineering programs (Mechanical Engineering, Electrical Engineering, and Civil Engineering) by instructors from the computer science department. The course is an introduction to computer systems, problem-solving methods, and algorithm development. Fundamental programming skills are taught using the C language, and the high-level programming language (with equation solvers) MATLAB is also used to develop the students’ capacity to solve complicated mathematical problems. The choice of language in using structured programming is taught in this course, including designing coding, debugging, documenting programs, and the use of good programming style, especially as applied to basic engineering applications. As indicated for some of the other universities (Table 1), second-level programming skills, such as numerical or computational methods, are normally offered at the junior level to harness student programming and problem-solving skills with a focus on solving advanced mathematical equations used in engineering. This type of course is currently not offered in the PMU Civil Engineering Program, although a curriculum upgrade in this respect is under consideration. This upgrade would be implemented as a standalone course or as an additional topic inserted into one of the civil engineering courses. With limited “space” in the curriculum, another computing-based course could be added as a technical elective, since the other program within the college of engineering (i.e., mechanical engineering) offers a mandatory computational method course. Second-level programming skills in the undergraduate curriculum are normally required for students who intend to pursue graduate study. However, it is still premature for undergraduate seniors to develop their own coding for solving structural analysis problems via the numerical finite element method, unless there is an advanced

course for computer-based finite element methods offered as a technical elective. A master's degree program under the Civil Engineering Department is under development and is expected to be launched within the next academic year. This master's degree program could offer computing courses, including Advanced Numerical Methods for Civil Engineering and Computer Applications in Civil Engineering, as technical electives for advanced undergraduate courses.

In the Senior Design I and II (capstone) courses, students are required to use applicable modern engineering software in their engineering projects as part of achieving ABET student learning outcome 7: *an ability to acquire and apply knowledge as needed using appropriate learning strategies* (Chandwani et al., 2013). Students can learn more than one modern engineering software program to complete their project, as the capstone courses must cover at least two major basic design courses. Again, the utilization of software in these courses is intended to develop a critical comparison against conventional or standard-based design procedures. Students are not expected to just accept the computer output without validating it using available data obtained from either experimental work or standard analytical techniques. In building their design project, for example, students need to perform comparative design outcomes between a code-based (standard) design and computer output, and they must also conduct an iterative design process to obtain an optimum design (Chowdhury et al., 2020).

With increased utilization of commercial software packages, civil engineering graduates are increasingly expected to possess a certain degree of expertise in computing skills and are at the same time required to be aware of professional misconduct resulting from computer misuse. In applying computer software for engineering design, students during their studies are trained to avoid the so-called “black box syndrome”—i.e., trusting the computer outputs or computer solutions without proper justifications or validations (Hu & Zhou, 2021). To mitigate this concern, the application of computer software packages should not be taught separately from the main course. Students must develop strong skills first in analytical problem solving and must be able to assess computer output using available references or analytical solutions. Offering an independent course in software package application would create a sense of “instant expertise” without adequate fundamental knowledge behind the software creation. Students must be trained to have a sense of confidence in their analytical judgment and to be critical of computer-based solutions.

It is worth noting that in practice, civil engineers work on the three-dimensional design of civil structures, such as municipal buildings, and therefore require the assistance of effective civil engineering software. These tools are particularly useful for the design of models of buildings, dams, bridges, railways, roads, highways, airports, and all aspects of city engineering. Such tools are known to help civil engineering with all design and construction requirements, e.g., design, visualization, and analysis. In addition, these tools can help civil engineers finalize infrastructure costs as well as provide resources and planning. For advanced requirements, the tools can be combined with versatile CAD software (Gifford, 1987; Grigg et al., 2004; Grigg et al., 2005; Vergara et al., 2015). However, according to Table 1 (Part B), only two to three computer-based programming courses are offered for civil engineering graduate students in almost all universities (either locally or in the US). Consequently, during their studies, students should be trained on some relevant commercial software. As mentioned previously, training on the application of computer software packages should be taught in parallel with the main course to develop confidence and adequate analytical skills alongside strong reasoning skills with respect to computer output.

Table 1(Part B). (Universities in the United States): Computing-based courses in civil engineering programs.

No	University and program name	Credit hours for the degree	Computing course name (Credit hours)	Course prerequisites	Course description
1	The City College of New York – Civil Engineering	136	1.CSc 10200 Introduction to Computing (2)	Math 19500 or Pre/Co-requisite Math 20100	The structure and operation of a computer, concepts and properties of an algorithm and a programming language. Introduction to programming in a modern programming language, such as C/C++. Emphasis is placed on applications of interest to scientists and engineers
2	Massachusetts Institute of Technology	180	2. CE 31500 Computational Methods in CE (3)	Math 39100, CE 26300 and CE 2310, CSc 10200	Algorithmic formulation of the solution to civil engineering problems, flow-charts, and solutions to algebraic and differential equations common to civil engineering
3	Georgia Tech School of Civil and Environmental Engineering	128	1. CI-M 1.00 Engineering Computation and Data Science 2. CI-M 1.000 Introduction to Computer Programming	Calculus 1 (GIR) Coreq: 18.03	Engineering problems in a computational setting, such as data analysis, filtering, regression, machine-learning models, programming The fundamentals of computing, computer programming, use of MATLAB programming
4	Purdue University –West Lafayette	132	1. CCS1371 Computing for Engineers (3) 2. CEE 1770 Introduction to Engineering Graphics and Visualization (3)	Core B AE and ME 1770	Foundations of computing with analysis of algorithms and programs for engineering problem solving. Engineering graphics and visualization, sketching, drawing, and solid modeling
5	Stanford University –Civil and Environmental Engineering	180	1. CGT 16400Computer Graphics 1. CME 100 Computational Methods for Engineers (3) 2.CEE101DComputations or Science and Engineering (3) 3. CEE101D* Computation Civil and Environmental and Engineering (3)	None MATH 3941 Sophomore year Sophomore year	Introduction to computer graphics documentation for civil engineering and construction-related professions Mathematical calculations using programming software Science and engineering problem solving by using software lab Science and engineering problem solving by using software course
6	University of Illinois – Urbana Champaign	128	1. SE 101 Engineering Graphics and Design (3) 2. CS 101	One lab and one lecture section is compulsory Math 220	Computer-aided design software modeling of parts and assemblies Fundamental principles, concepts, and methods of

			Introduction to Computing, Engineering, and Science (3)		computing
7	The University of Texas	124	1. CE 311 K (3)	Math 408D	Organization and programming of civil engineering problems for computer solutions
8	Virginia Tech University	125	1. CEE 2824 Civil Engineering Drawings and CAD (3) 2. CEE 2834 Civil Engineering Drawings and Virtual Modeling (3)	CEE 2814 Interdisciplinary programs	CAD software in civil engineering, construction, and land development projects CAD, BIM, and GIS software in civil engineering
9	University of Michigan – Civil and Environmental Engineering	131	1.CEE 4020 Computer Applications: Visualizing and Communicating Design Information (3) 2. CEE 4760 Optimization Methods in Civil and Environmental Engineering (3)	CEE 3332 or CEE 3401(C) MA 2160 and MA 2320 or MA 2321 or MA 2330	Problem solving using industry standard software, such as Civil3D, is applied to civil and environmental engineering projects, such as terrain modeling, earth work calculations, and road alignment. Concepts involving data management, data visualization, and risk analysis are introduced Decision analysis and optimization techniques, including linear programming, nonlinear programming, and dynamic programming. Computer-based solutions of design problems in various civil and environmental engineering specialty areas are considered
10	Carnegie Mellon University	128	1. 99-101 Computing @ Carnegie Mellon (3) 2. 12-271 Introduction to Computer Application	None None	Developing foundational computing and information literacy skills Introduction to use of computer-based applications in civil engineering

However, the choice of software may not be that simple, as the market is overflowing with such software (Abudayyeh et al., 2006; Chowdhury et al., 2020). For convenience, some of the best civil engineering software programs are included in Table 2. Among others, for example, GeoStudio's integrated products enable one to work across a broad range of engineering use cases, such as dams and levees, reinforced walls and slopes, excavations and open pit mines, roads, bridges and embankments, environmental protection, groundwater, ground freezing and climate change, earthquake deformations, and vadose zone hydrology. AutoCAD software greatly facilitates the design of 2D and 3D models. It helps to model and visualize 3D, and allows working with construction designs. AutoCAD provides 3D digitizing and point displacement, 3D navigation, and smart dimensions, as well as 2D and 3D drawing and annotation functions. STAAD Pro is another design program developed to perform a complete analysis and design of any structure much faster than before. Basically, it facilitates the BIM workflow by using a physical prototype that automatically converts to an analytical prototype of the structural anatomy. Moreover, Primavera provides powerful tools to help in the planning, building, and operation of large assets, including complex and exclusive day-to-day and replicable projects. These projects draw on industry expertise in manufacturing, infrastructure, trade, and other areas. It assists in the management of projects and service and program portfolios, and it allows planning activities and tasks on different programs or portfolios.

Table 2. Computing skills for civil engineering technical courses

No	Course code and name (credit hours)	Computing skills			Notes about software used
		Basic	Programming (coding)	Applying software for engineering design and analysis	
1	GEEN 1211 Introduction to Engineering (2)	x			Report writing, presentation, spreadsheets with Excel
2	GEEN 2311 Engineering Mechanics I: Statics (3)				
3	GEEN 2211 Engineering Computing (2)		x		C and MATLAB
4	GEEN 2312 Engineering Mechanics II: Dynamics (3)				
5	CVEN 2313 CAD for Civil Engineering (3)			x	AutoCAD
6	GEEN 2313 Thermodynamics (3)				
7	MEEN 2313 Solid Mechanics (3)			x	Solidworks
8	GEEN 3311 Introduction to Fluid Mechanics (3)				
9	CVEN 3322 Materials in Civil Engineering (3)	x			Spreadsheet, report writing
10	CVEN 3311 Structural Analysis (3)			x	Structural analysis software SAP2000
11	CVEN 3323 Engineering Geology (3)				GeoStudio (Geoslope)
12	CVEN 3341 Engineering Measurement (3)	x		x	Spreadsheet, lab report writing, GIS software
13	CVEN 3331 Environmental Engineering Fundamental (3)				GeoStudio
14	CVEN 3343 Engineering Probability and Statistics (3)	x		x	DPSS, Minitab, Excel, statistical software
15	CVEN 3312 Reinforced Concrete Design (3)			x	Structural analysis software ETABS
16	CVEN 3344 Sustainable Engineering (3)	x			Spreadsheets, report writing
17	CVEN 3332 Hydraulic Engineering (3)	x			GeoStudio, spreadsheets, lab report writing
17	CVEN 3301 Internship (3)	x			Report writing and

				presentation
18	CVEN 4313 Design of Steel Structures (3)		x	Structural analysis software SAP2000
19	CVEN 4323 Introduction to Geotechnical Engineering (3)	x	x	Geoslope, spreadsheet, lab report writing
20	CVEN 4396 Senior Design I (3)	x	x	Applicable software for design
21	CVEN 4342 Transportation Engineering (3)		x	Pavexpress, AASHTO are Pavement ME Design
22	GEEN 4311 Engineering Economy (3)		x	Solver software
23	CVEN 4314 Construction Management (3)		x	MS Project, Primavera
24	CVEN 4324 Foundation Analysis and Design (3)		x	STAAD, GEO5, spreadsheet programming
25	CVEN 4334 Air Pollution and Control (3)	x		
26	CVEN 4333 Water and Wastewater Treatment (3)	x		
27	CVEN 4397 Senior Design II (3)	x	x	Applicable software for design

6. Conclusion

In general, it can be concluded that computing competency in the civil engineering discipline can be adequately addressed by offering explicit computing-based courses in combination with technical courses that use the applications of basic or advanced software in engineering design. Explicit computing-based courses include courses with computer science-based knowledge, such as programming (coding) using applicable computer languages. Two levels of programming courses can be offered in a civil engineering program: introduction to programming and numerical or computational methods. As demonstrated by the majority of civil engineering programs, it is very challenging to insert another explicit computing-based course in the curriculum. One of the solutions to overcome this issue is to offer technical elective utilizing existing courses available from other programs. In this way, the curriculum will give students who have a high degree of interest in advancing computing and programming skills to the next level the opportunity to do so. The other way to improve computing competency is to upgrade learning outcomes and the associated key performance indicators at both the program and course levels. The specialized use of computer software in engineering design can always be incorporated into certain courses within the curriculum.

Acknowledgment

The authors would like to acknowledge contributions from PMU senior civil engineering students for their verbal discussion and sharing of experiences in the development of computing skills.

References

- Abudayyeh, O., Dibert-DeYoung, A., Rasdorf, W., & Melhem, H. (2006). Research publication trends and topics in computing in civil engineering. *Journal of Computing in Civil Engineering*, 20(1), 2-12. [https://doi.org/10.1061/\(ASCE\)0887-3801\(2006\)20:1\(2\)](https://doi.org/10.1061/(ASCE)0887-3801(2006)20:1(2))
- Asif, M., Bajpai, S., & Kumar, M. (2015). Role of simulation in civil engineering education in India. *Journal of Civil and Construction Engineering*, 1(2).
- Asmaa, G., Salih, H., & Ahmed, A. (2014). The effective contribution of software applications in various disciplines of civil engineering. *International Journal of Civil Engineering and Technology*, 5(12), 316-333.

- Ayadat, T., Ahmed, D., Chowdhury, S., & Asiz, A. (2020). Measurable performance indicators of student learning outcomes: A case study. *Global Journal of Engineering Education*, 22(1), 40-50. Al Khobar Kingdom of Saudi Arabia.
- Baker, N. C., & Rix, G. J. (1992). Computing in civil engineering: Current trend and future directions. *Journal of Professional Issues in Engineering Education and Practice*, 118(2), 139-155. [https://doi.org/10.1061/\(ASCE\)1052-3928\(1992\)118:2\(139\)](https://doi.org/10.1061/(ASCE)1052-3928(1992)118:2(139))
- Bowen, J. (2004, June). Motivating civil engineering students to learn computer programming with a structural design project. Paper presented at 2004 Annual Conference, Salt Lake City, Utah. 10.18260/1-2—13064. <https://doi.org/10.13140/2.1.3458.7208>
- Chandwani, V., Agrawal, V., & Nagar, R. (2013). Applications of soft computing in civil engineering: A review. *International Journal of Computer Applications*, 81(10). <https://doi.org/10.5120/14047-2210>
- Chowdhury, S., Ayadat, T., & Asiz, A. (2020). Rubric-based scoring for engineering senior design course assessment and grading. *World Transactions on Engineering and Technology Education*, 18(4), 1-10.
- Clough, R. W. (1958). Structural analysis by means of a matrix algebra program, *Proceedings of the first conference on electronic computation*. 109-132, ASCE, New York.
- Dai, Q., & You, Z. (2006, June). Using Matlab to solve engineering problems for undergraduates. Paper presented at 2006 Annual Conference & Exposition, Chicago, Illinois. 10.18260/1-2—973.
- Danso, H. (2013). Construction student's awareness of CAD programs of Polytechnics in Ghana. *Global Journal of Engineering, Design and Technology*, 2(3).
- Ehsan, M. (2019). Feasibility of soft computing in civil engineering – A delusion. *Third international conference on applied researches in structural engineering and construction management*. Sharif University of Technology, Tehran, Iran. <https://doi.org/10.2174/1875323X01002010018>
- Gerber, D. J., Khashe, S., & Smith, I. F. C. (2015). Surveying the evolution of computing in architecture, engineering, and construction education. *Journal of Computing in Civil Engineering*, 29(5). [https://doi.org/10.1061/\(ASCE\)CP.1943-5487.0000361](https://doi.org/10.1061/(ASCE)CP.1943-5487.0000361)
- Gifford, J. B. (1987). Microcomputers in civil engineering: Use and misuse. *Journal of Computing in Civil Engineering*, 1(1), 61-68. [https://doi.org/10.1061/\(ASCE\)0887-3801\(1987\)1:1\(61\)](https://doi.org/10.1061/(ASCE)0887-3801(1987)1:1(61))
- Grigg, N. S., Criswell, M. E., Fontane, D. G., & Saito, L. (2004). Integrated civil engineering curriculum: Five-year review. *Journal of Professional Issues in Engineering Education and Practice*, 130(3). [https://doi.org/10.1061/\(ASCE\)1052-3928\(2004\)130:3\(160\)](https://doi.org/10.1061/(ASCE)1052-3928(2004)130:3(160))
- Grigg, N. S., Criswell, M. E., Fontane, D. G., & Siller, T. J. (2005). Information technology in civil engineering curriculum. *Journal of Professional Issues in Engineering Education and Practice*, 131(1). [https://doi.org/10.1061/\(ASCE\)1052-3928\(2005\)131:1\(26\)](https://doi.org/10.1061/(ASCE)1052-3928(2005)131:1(26))
- Kheyfets, A. L., & Vasilieva, V. N. (2017). 3D modeling as method for construction and analysis of graphic objects. *Conference Series: Materials Science and Engineering*. Chelyabinsk, Russia. <https://doi.org/10.1088/1757-899X/262/1/012104>
- List, G. (2007). AC 2007-1478: Introducing civil engineering analysis through programming. *American Society of Engineering Education*, 12.961.1. <https://doi.org/10.18260/1-2--2303>
- Osama, A., Cai, H., Fenves, S. J., Law, K., O'Neill, R., & Rasdorf, W. (2004). Assessment of the computing component of civil engineering education. *Journal of Computing in Civil Engineering*, 18(3), 187-195. [https://doi.org/10.1061/\(ASCE\)0887-3801\(2004\)18:3\(187\)](https://doi.org/10.1061/(ASCE)0887-3801(2004)18:3(187))
- Pengzhen, L., Shengyong, C., & Yujun, Z. (2012). Artificial intelligence in civil engineering. *Mathematical Problems in Engineering*. <http://doi.org/10.1155/2012/145974>
- Petuhova, A. (2015). Engineering graphics course using modern software systems for students of civil engineering university, 3(1), 47-58. <https://doi.org/10.12737/10458>
- Rasdorf, W. J. (1985). Computer programming in the civil engineering curriculum. *Journal of Professional Issues in Engineering*, 111(4). [https://doi.org/10.1061/\(ASCE\)1052-3928\(1985\)111:4\(141\)](https://doi.org/10.1061/(ASCE)1052-3928(1985)111:4(141))
- Sampaio, A. Z., Henriques, P. G., & Martins, O. P. (2010). Virtual reality technology used in civil engineering education. *The Open Virtual Reality Journal*, 2, 18-25. <https://doi.org/10.2174/1875323X01002010018>

- Sampaio, A. Z. (2017). BIM as a computer-aided design methodology in civil engineering. *Journal of Software Engineering and Applications*, 10, 194-210. <https://doi.org/10.4236/jsea.2017.102012>
- Shinde. A. J., Salunkhe, M. D., & Navale, N. V. (2018). Computer applications in civil engineering software in civil engineering. *International Journal of Advance Engineering and Research Development*, 5(4).
- Tavakoli, A., & Klika, K. L. (1991). Construction management with AutoCAD. *Journal of Management in Engineering*, 7(3). [https://doi.org/10.1061/\(ASCE\)9742-597X\(1991\)7:3\(267\)](https://doi.org/10.1061/(ASCE)9742-597X(1991)7:3(267))
- Vergara, C. E., Cavanaugh, M., Sivakumar, S., Briedis, D., Dionise, T. D., Sticklen, J., & Urban-Lurain, M. (2015). Towards a framework for assessing computational competencies for engineering undergraduate students, *122nd ASEE annual conference and exposition*. ASEE Michigan. <https://doi.org/10.18260/p.24925>
- Wangping, S. (2011). AC 2011-308: Teaching computer-programming skills to engineering and technology students with a modular programming strategy. *American Society of Engineering Education*. <https://doi.org/10.18260/1-2-18625>
- Wilson, E. L. (1963). *Finite element analysis of two-dimensional structures* [Doctoral dissertation, University of California at Berkeley]. University of California at Berkeley, California.
- Xiaowen, H., & Zhou, Q. (2021). MATLAB software in the numerical calculation of civil engineering, *The 2020 international conference on machine learning and big data analytic for IoT security and privacy*, 730-734. <https://doi.org/10.1007/978-3-030-62746-1-11>
- Zhou, L. (2016). Application of civil engineering in computer. *Proceedings of the 2016 international conference on education, management, computer and society*. <https://doi.org/10.2991/emcs-16.2016.434>

Copyrights

Copyright for this article is retained by the author(s), with first publication rights granted to the journal.

This is an open-access article distributed under the terms and conditions of the Creative Commons Attribution license (<http://creativecommons.org/licenses/by/4.0/>).