

Project Description

CONTENTS

1. Introduction	1
2. Results from Prior NSF Support	2
3. Intellectual Merit	4
4. Broader Impacts	10
5. Partnerships and External Advisory Board	12
6. Assessment	13
7. Timeline	14

1. INTRODUCTION

Computational scientists typically draw upon knowledge from several disciplines, including computer science, mathematical science, and natural science. Realizing the full potential of computational science for discovery requires interdisciplinary teams whose members can take full advantage of advanced hardware architectures and software environments, in addition to mathematical modeling and algorithm development. To conduct effective research, computational scientists should exhibit

- Depth within their chosen disciplines plus breadth across relevant disciplines, overcoming a silo mentality, and
- Experience with advanced cyberinfrastructure (CI), including some professional practices required for effective large-scale computation.

The newly established Center for Interdisciplinary Scientific Computation (CISC) <http://cos.iit.edu/cisc> at Illinois Institute of Technology (IIT or Illinois Tech) is proposing several new initiatives to educate computational scientists with qualities highlighted above. We will educate high school through doctoral students. Learning will be curricular and extra-curricular. We will partner with nearby national laboratories, companies engaged in advanced computing, and schools whose students have less access to research experience and high-end computing facilities.

The Challenges. Computing solutions to complex scientific problems require properly educated computational scientists. There are two kinds of challenges that must be addressed in providing this proper education.

Academic Silos. Computational science draws on multiple disciplines, including computer science, mathematical science, and natural science. Students must gain a deep understanding of their chosen discipline. Computational scientists should avoid becoming a jack of all trades and a master of none.

- Computer scientists must understand how emerging languages and architectures enable faster and more scalable computations, and they must be prepared to make the next advances.
- Computational mathematicians and statisticians must be able to describe and analyze truncation and round-off errors in numerical algorithms, as well as measurement and misspecification errors of statistical models. They must know how to construct the most efficient algorithms for different kinds of problems.
- Computational biologists, chemists, and physicists must grasp the important scientific ideas that need to be captured by computations. They must understand the benefits and limitations of using computer modeling in their respective disciplines and grasp the next opportunities for scientific advancement.

But this is not enough! Computational scientists should be a master of one trade and conversant with others. They must develop cross-disciplinary perspectives that will make them effective members of interdisciplinary teams tackling large, complex computational science problems.

Ignorance of Good Practice for Large-Scale Computation. Solving problems using advanced CI deviates in important ways from what students typically learn in their coursework. See [42, Sect. 2.4] for a more detailed discussion. Here are some highlights:

- Effective algorithms that take advantage of multi-core, distributed memory architectures may be substantially different than algorithms designed for a single CPU with only one core.
- The software required to solve large problems is often drawn from multiple sources, written in several languages, and developed over the course of years, by teams of experts, e.g., PETSc [2].
- Those who perform computational experiments and those who demonstrate the performance of their new algorithms must ensure that their results are reliable and can be reproduced by others [39].
- Software contributors must ensure that their software is robust, interacts well with other software, and can be extended by those who come afterwards. See [5] for a recent effort to promote these principles broadly.

We propose to overcome these challenges in educating the next generation of computational scientists.

Why CISC is Poised to Lead This. CISC was created in May 2017 to leverage Illinois Tech’s existing strengths in computational science for greater impact. CISC has office space, a modest budget, and a 256-core cluster (see the section on Facilities, Equipment and Other Resources). We also have the good will of supporters among faculty and administrators. CISC has initiated a series of “matchmaking” seminars that introduce computational scientists to each others’ research. The aim is to promote new interdisciplinary computational research groups that attract external funding. In support of this aim, CISC also sponsored a seed grant competition in Fall 2017. Fred J. Hickernell, PI and CISC’s director, is one of several computational mathematicians at Illinois Tech and previously served for twelve years as chair of the applied mathematics department. Our computer science department has a historically strong group in high performance computing led by co-PI Xian-He Sun. In recent years, the biology, chemistry, and physics departments have hired computational scientists, including co-PIs David Minh (chemistry, CISC’s associate director) and Jeff Wereszczynski (physics). Co-PI Sou-Cheng Choi is a lead researcher in machine learning at Allstate Insurance Company’s Automotive and Life/Retirement innovation teams and also a research associate professor in applied mathematics at Illinois Tech.

In addition to the (co-)PIs, we have enlisted two senior personnel: Norman Lederman, who has a long and distinguished record in science education, and Kiah Wah Ong, an early-career colleague with strong teaching experience.

The (co-)PIs have experience developing curricula, mentoring high school students through PhD students, and partnering with College of DuPage, Argonne National Laboratory, and Fermilab. But our past record has been working primarily independently. The creation of CISC and the resources provided by Illinois Tech have provided us a platform with the potential to create a pipeline of computational scientists which starts at the high school level and extends to the doctoral level. The resources requested in this proposal will make that potential a reality by enabling the creation of a novel, strong, multi-faceted training program for cross-disciplinary computational scientists.

2. RESULTS FROM PRIOR NSF SUPPORT

2.1. Experience of Hickernell and Choi. NSF-DMS-1522687, *Stable, Efficient, Adaptive Algorithms for Approximation and Integration*, \$270,000, August 2015 – July 2018. Hickernell is PI and Choi is senior personnel.

2.1.1. Intellectual Merit. One of the primary outcomes of this project in relation to the present proposal is the development of the Guaranteed Automatic Integration Library (GAIL) [7]. This library comprises univariate and multivariate integration, as well as univariate function approximation and

optimization algorithms that automatically determine the sample size required to meet user-defined error tolerances. The library does not rely on interval arithmetic, as is done in INTLAB [35, 43, 44]; but like INTLAB, GAIL comes with theoretical guarantees that common adaptive algorithms lack. The recent GAIL developments include locally adaptive function approximation and optimization [6, 9], adaptive quasi-Monte Carlo cubature [18, 23], and the ability to set a hybrid error tolerance involving both absolute and relative error criteria [20]. Other articles, theses, software, and preprints supported in part by this grant include [1, 11, 22, 19, 17, 24, 28, 29, 31, 33, 34, 41, 48, 51, 52, 53].

2.1.2. Broader Impacts. Three students (two female) have completed their PhD degrees, and three students are in the midst of their PhD studies. One PhD graduate spent time as a student working at Fermilab helping physicists implement modern Monte Carlo methods. Another PhD student is picking up where the first one left off. Three students (two female) completed their MS theses. More than a dozen undergraduate students have been mentored (primarily during the summer), over the course of three summers. Some were supported by this grant, while most were supported by other funding sources. Hickernell has embedded the new adaptive (quasi-)Monte Carlo research in his graduate Monte Carlo course, and some students in that course have contributed to *GAIL*. Hickernell, Choi, and their students have written an encyclopedia article; given numerous conference and colloquium talks; organized a conference and arranged special conference sessions at multiple conferences. Hickernell has given an invited conference tutorial and is one of the program leaders for this year’s SAMSI program on quasi-Monte Carlo sampling. Hickernell received the 2016 Joseph F. Traub Prize for Achievement in Information-Based Complexity.

2.2. Experience of Sun. NSF-CNS-1162540, *CSR: Medium: Collaborative Research: Decoupled Execution Paradigm for Data-Intensive High-End Computing*, \$842,298 (IIT share \$280,766), June 2012 – August 2016. Sun is the PI.

2.2.1. Intellectual Merit. This collaborative project developed a decoupled execution paradigm (DEP) to address input/output (I/O) bottleneck issues. This paradigm enables users to identify and handle data-intensive operations separately.

2.2.2. Broader Impacts. Technical hurdles have been identified for decoupled execution, spanning system architecture, programming model, and runtime system. Results are available as recent publications [12, 13, 16, 14, 15, 27, 45, 49, 50] and other I/O related publications. In addition, in the past five years, Dr. Sun has graduated 3 PhD and 11 masters students, and supervised 5 postdoctoral and 4 graduate researchers.

2.3. Experience of Wereszczynski. PI of NSF-MCB-1552743 “CAREER: The Effects of Post-translational Modifications and Histone Variants on Chromatin Fiber Dynamics” June 2016 – May 2021, \$790,129.

2.3.1. Intellectual Merit. The aim of this CAREER award is to use and develop multiscale biophysical simulation techniques to study how DNA is compacted in the cell, and how these mechanisms are regulated to affect gene expression. The PI’s group addresses these issues through the use of molecular dynamics (MD) simulations, along with collaborations at Argonne National Laboratory and the University of Iowa. Early results have been presented at the annual Biophysical Society Meeting in 2017 and 2018, and a manuscript detailing the mechanisms by which post-translational modifications to the “H3 tails” affect the structures and energetics of nucleosomes is under minor review at *eLife* [36].

2.3.2. Broader Impacts. To date, this proposal has funded four trainees: one postdoctoral scholar, one PhD student, and two undergraduates from the College of DuPage. Both undergraduates performed 10-week summer research projects, during which time they learned how to set up, perform, and analyze MD simulations of various biomolecular systems. Robert Hickok, who was in our lab

in the summer of 2016, is currently enrolled at the University of Illinois at Chicago, whereas Meet Patel, who worked with us in 2017, is enrolled at the Georgia Institute of Technology.

3. INTELLECTUAL MERIT

3.1. Skills to Be Learned. Our primary emphasis is educating CI contributors (CICs), and our secondary emphasis is educating CI users (CIUs). Our goal is to train students from high school through PhD level in skills of various difficulty, some building upon others. These include the skills in the table below. The sections describing our initiatives where these skills will be learned are listed in the far right column.

Learning Outcomes	CIU	CIC	Learned
<i>Use</i>			
1. Write and run numerical programs on a single CPU	HS	HS	3.3, 3.4.1, 3.5
2. Run numerical programs that utilize multiple cores and/or a GPU on a single machine	UG	HS	3.3, 3.4.1, 3.4.3, 3.4.4, 3.5
3. Run jobs that require more than one node with tight connectivity	G	UG	3.4.2, 3.4.4, 3.4.5, 3.6
4. Run jobs on a top-500 supercomputer	G	UG	3.4.2, 3.4.4, 3.5, 3.6
5. Use tools for analysis and visualization of large-scale simulations	G	G	3.4.2, 3.4.5, 3.6
6. Solve scientific problems that require multiple libraries and languages	G	UG	3.4.3, 3.4.5, 3.6
7. Execute reproducible scientific computations	UG	UG	3.4.2, 3.4.3
<i>Develop & Optimize</i>			
8. Write numerical programs that take advantage of multiple cores and/or a GPU on a single machine	UG	HS	3.3, 3.4.2, 3.4.3, 3.4.4
9. Make additions to a well-documented numerical software library consisting of documented, tested, robust routines		G	3.4.3, 3.6
10. Contribute to a numerical software library that takes advantage of high performance computing architectures		G	3.4.5, 3.6
11. Analyze the computational efficiency of individual algorithms and identify performance bottlenecks		UG	3.4.1, 3.4.2
12. Analyze the computational efficiency of large-scale simulations and identify performance bottlenecks		G	3.4.5, 3.6
<i>Apply</i>			
13. Evaluate whether simulation output accurately reflects the natural phenomenon it is designed to emulate	UG	UG	3.4.1, 3.4.2, 3.6
14. Appreciate how a computational result informs decision-making in the application domain and its effect on what kind of computational result is needed	G	G	3.4.1, 3.4.2, 3.6
15. Know how different modeling assumptions in the application domain are tied to the choice of different computational methods and computing environments	G	G	3.4.2, 3.5, 3.6
16. Collaborate with computational scientists outside their own majors	UG	UG	3.4.1, 3.6

Many of the learning outcomes above parallel to those provided in Table 1 of [42]. However, that table focuses on the outcome for a PhD in computational science. Our table above provides intermediate outcomes for those in high school, bachelor’s, and master’s study. A future CIC should ideally develop all of the skills labeled HS in high school, all of the skills labeled UG during undergraduate studies, and all of the skills labeled G during graduate studies. The analogous interpretation applies to CIUs. Our proposed innovations, outlined in the remainder of this section, provide opportunities to future CICs and CIUs to develop these skills.

The degree of mastery of any particular skill may depend on whether the computational scientist is majoring in computer science, mathematical science, or natural science. Moreover, our students may only be with us for part of their high school through PhD studies. We are ready to help those whose preparation is lacking catch up to where they need to be. When students leave our program, they should be fully prepared for the next steps in their growth as computational scientists.

3.2. Proposed Innovations. Our innovations will better prepare CICs and CIUs via several modes of learning for different kinds of students. They can be summarized as follows:

Section	Innovation	Target	
3.3	Summer computational science course	CIU	CIC
3.4.1	Enriched existing computational science offerings	CIU	CIC
3.4.2	Graduate theses incorporating advanced CI	CIU	CIC
3.4.3	Professional practices for computational science course		CIC
3.4.4	Undergraduate parallel and distributed computing course		CIC
3.4.5	Large-scale scientific computation course		CIC
3.5	Research experiences for community college students	CIU	CIC
3.6	CISC undergraduate and graduate summer fellowships		CIC

Several of these innovations involve permanent expansions of our curriculum. The others are extra-curricular activities designed to complement what is learned in the classroom. We have a track record in most of these areas, but we want to go far beyond our past experience to establish something new and substantial.

3.3. Summer computational science course. Since 2013, Illinois Tech’s College of Science has run a (non-residential) three-week computational science summer course for Chicago area high school students. The goal has been to introduce them to solving mathematical and scientific problems using Mathematica. The Undergraduate Admissions Office publicizes and recruits students for the course, which has had an average enrollment of 9–10 students in the past. Starting in the summer of 2018, this summer course has an academic home in CISC, while continuing to partner with the Undergraduate Admissions Office, and in particular, Ms. April Welch. Dr. Kiah Wah Ong is the instructor.

We will broaden this course to include computing on the CISC cluster to introduce these high school students to multi-core computing (Skills 1, 2, and 8). Mathematica has the capability to perform computations in parallel, and we will also investigate the use of other languages, such as Python. This summer course will encourage the students to pursue degrees involving high performance computation and better prepare them for potential careers as computational scientists. We also hope to attract the students to Illinois Tech’s computational science offerings.

Because our students are expected to have diverse backgrounds in computing, we will begin the course by assessing their existing skills and then break them into groups so that those who need to learn more fundamental skills may do so, while those with more advanced knowledge can be given more challenging assignments. Illinois Tech students will serve as teaching assistants (TAs) for the course, which will provide them with the opportunity to reinforce their own learning by teaching the computational science that they have learned.

The timing of this course will overlap with the summer research experiences for community college students (Sect. 3.5) and the research experiences of the CISC fellows (Sect. 3.6). We will inspire the students in this summer course by introducing them to the work of those who are a few steps ahead of them in their development as computational scientists.

Our goal is to host 15 high school students in the summer of 2019 and increase the number each year to 30 high school students steady state in the summer of 2021. The summer course has been and will continue to be funded by tuition fees. In order to broaden access, we plan to offer tuition scholarships to those with financial need. We will also track their career trajectories through annual surveys.

To attain these goals, we will create hard- and soft-copy fliers describing our revamped computational science course. Ms. Welch’s colleagues in the Undergraduate Admissions Office will help us promote the course as they visit Chicagoland high schools. Hickernell and Ong will make themselves available to brief admissions counselors about the course and tell visiting high school students about it.

We also plan to open this summer computational science course to undergraduate students, primarily from Illinois Tech. Our goal is to attract 5–10 such students per year. Many of our students have scholarships or financial aid, which covers the summer term as well. We expect students who may not have a strong computing background—but have an curiosity in knowing more—will be attracted to this course. This will provide them a stepping stone to our more advanced computational science offerings.

3.4. Strengthened undergraduate and graduate level computational science coursework. Illinois Tech already has a substantial computational science curriculum. However, it has several deficiencies, which we aim to address.

CISC will post fliers around campus and publish pages on the CISC website describing our existing computational science offerings as well as our new and expanded offerings described below. Each semester we will broadcast information about the courses being offered to Illinois Tech students and their advisors to recruit more and better students into these classes.

3.4.1. Enriched existing computational science offerings. Illinois Tech offers quite a number of computational science courses. Computer science offers high performance computing at the undergraduate and graduate levels. Applied mathematics BS through PhD students are required to take at least one computational mathematics course and may also choose from a rich variety of computational mathematics electives. There are two undergraduate computational physics courses, a computational chemistry course, and a computational biology course.

These courses teach Skills 1, 11, and 13. However, students tend to take only the courses in their own major. Thus an applied mathematics student may not really have the chance to develop Skill 13. The (co-)PIs, who teach some of these courses, will liaise with other course instructors and the host departments to encourage non-majors to take these courses (Skill 16). We will explore how we might lower the pre-requisite barriers to help students break into new silos. We also will explore introducing some parallel computing (Skill 2) into these courses.

Minh has been playing a key role in developing the Bachelor of Science in Computational Chemistry and Biochemistry at Illinois Tech. To our knowledge, it is the first program of its kind in the country. In addition to a standard Bachelor of Science curriculum certified by the American Chemical Society, the recently introduced degree program requires courses from computer science and three specialized courses: computational quantum chemistry, computational biochemistry and drug design, and cheminformatics. The former has been offered on an annual basis and the latter two will be developed by Minh. Minh will develop the courses in a way that does not require advanced knowledge of physical chemistry and is therefore maximally inclusive to different majors.

The Monte Carlo methods graduate course taught by Hickernell every fall requires every student to complete a course project. Each year since 2015, **HAVI**, a global logistics company, has offered

a simulation project for this class that arises from their practical research. Two to three groups of five students each have dealt with the complexities of understanding data, formulating appropriate mathematical models to answer the question posed by HAVI, settling on a suitable simulation platform, performing and refining their simulations, and reporting their results to HAVI. These projects have gone well beyond the simple problems used to teach fundamental Monte Carlo concepts and have given students an added dimension of industrial research to their education. The approaches and the insights provided by the students have impressed the HAVI researchers.

We propose to forge additional partnerships between local industry and our computational courses, which will help our students acquire Skill 14. This may take the form of course projects, guest lectures, and/or internship opportunities. As was the case with HAVI, we will harness our alumni connections. We will also make cold calls on companies performing extensive computation in the course of their work.

3.4.2. *Graduate theses incorporating advanced CI.* With the support of the other initiatives described above and below, we will be able to direct our graduate students towards more meaningful MS and PhD thesis topics that rely upon large-scale computation. This will help them learn Skills 3, 4, 5, 7, 8, 11, 13, 14, and 15.

3.4.3. *Professional practices for computational science course.* A few years ago, Choi and Hickernell piloted a one-credit course, MATH 573 Reliable Mathematical Software, primarily for their own research group. The course covered truncation and round-off errors, stopping criteria for adaptive software, creating software collaboratively via Git and GitHub, documentation, input parsing and validation, and reproducible computation. The rationale for the course was feedback from those outside academia that our computational mathematics students, while very knowledgeable about numerical algorithms, lacked some of the skills required to implement these algorithms in an industrial or government research setting. Students completed a project consisting of creating a small software library or extending an existing library. Several of the students in this course participated in developing GAIL [7]. This experience helped one student land an academic job teaching computer science, three students land computationally oriented positions in industry, and a fifth student gain admission to a PhD program.

We will grow this course to a standard three-credit undergraduate/graduate elective course. In addition to a fuller treatment of the above topics, which address Skills 7–9, we will add material on working in a multi-lingual environment (Skill 6) and small-scale parallel computation (Skills 2 and 8). The prerequisite will be any undergraduate course that covers numerical computation, therefore making this course accessible to a wider audience.

3.4.4. *Undergraduate parallel and distributed computing course.* Sun will be responsible for enhancing CS451 Introduction to Parallel and Distributed Computing. This new class is designed for CS students to learn the basic concepts and skills of parallel and distributed computing systems (Skills 2–4, 8). Sun will lead the effort to extend and enhance this class to attract and benefit students from other disciplines of science and engineering. There are definitely aspects of computer systems that are important for all computational scientists to know, but it is not clear how much non-computer science students can absorb if they are placed in the same class as computer science students. This is a non-trivial task, given that computer science students will typically have a much stronger foundation than other students, and one wants to benefit all students in the class. Sun, Hickernell, and Choi will share their experiences in teaching this class and the above-mentioned one. The aim will be to adjust the content and pre-requisites of these classes—keeping in mind the diversity of the intended audiences—and deciding whether the educational objectives are best served by having one, two, or even three classes. The three faculty may also engage in guest lecturing in each other’s courses.

3.4.5. Large-scale scientific computation course. Since 2006 the computer science department has offered a graduate course, Advanced Scientific Computation, which covers Skills 3, 5, 6, 10, and 12. Dr. Hong Zhang from Argonne National Laboratory has taught this course to students majoring in computer science, mathematics, physics, and engineering. Students were exposed to cutting-edge analytic and algorithmic research projects and gained hands-on large-scale numerical programming experience. This course has been a launching point for many students into careers in computational science. Several of the course projects were integrated into the Portable Extensible Toolbox for Scientific Computing (PETSc) [2], benefiting the scientific community in large. At least eight students received summer internships at Argonne after taking her course, and three of her students eventually became post-doctoral researchers at Argonne.

In the past, this course has run only every other year. We plan to increase the frequency of this course to annually, serving a dozen or more students, by advertising its existence and benefits.

Students desiring to be CISC fellows (see Sect. 3.6) will be more competitive if they take one or more of the courses mentioned above in Sect. 3.4.3–3.4.5. We will use this as marketing point to recruit students into these courses.

3.5. Research experiences for community college students. To encourage the pursuit of bachelor’s and advanced degrees by community college students, Wereszczynski has partnered in the past with the College of DuPage (CoD) to recruit students from their associate degree programs to engage in a ten-week summer internship program in his research group. CoD, with more than 28,000 students, is the second largest provider of undergraduate education in the State of Illinois. Many students who wish to pursue a career requiring an undergraduate or graduate degree start their college studies at CoD because for financial reasons.

We propose to broaden this program to include several CISC associated research groups at Illinois Tech. Each year we will work with Prof. Tom Carter of the physics department to advertise and recruit students to this program, with a particular focus on underrepresented minorities. These students will be matched with CISC labs based on their research interests and career goals. Our goal is to inspire and prepare these students for careers as computational scientists.

In the first phase of their internships, Minh will teach the cohort of CoD students in a week-long intensive computing “crash course” that will include topics such as an introduction to Linux, programming in Python, and the use of high performance computing resources. We will request an educational allocation on NSF XSEDE for students to run short benchmark molecular dynamics simulations on different computing architectures, including single CPUs, multiple cores on a single node, parallel jobs across multiple nodes, graphical processing units, and Intel Xeon Phi. Simulations of proteins in water including dihydrofolate reductase (23558 atoms) and cellulose (408609 atoms) will be run with NAMD [40]. Based on the results, students will compare the speed and scalability of the algorithms on different architectures and for different system sizes.

For the remaining nine weeks, CoD students will work directly in IIT labs on various computational projects, such as high performance Monte Carlo methods in Choi and Hickernell’s group, binding free energy calculations in the Minh’s group, parallel programming and performance optimization in Sun’s group, and simulations of biomolecular complexes in Wereszczynski’s group. To foster discussions among the CoD students, weekly lunches will be organized. In addition, at the end of the summer a research symposium will be organized in which students will present a short presentation on their work. Following the summer period, we will track the career trajectories of these students with annual follow up surveys, and we will maintain mentoring relationship with these students as are appropriate.

We will aim for four CoD students in 2019, six in 2020, and six in 2021. Students will be provided \$5000 stipends. After the project ends, we hope to secure funding to continue from other sources, such as Research Experience for Undergraduate (REU) funding and REU add-on supplemental funding.

3.6. CISC undergraduate and graduate summer fellowships. To provide more opportunities for the practice of the advanced CI skills that we are teaching our Illinois Tech undergraduate and graduate students, we plan to offer summer fellowships. These will be awarded on a competitive basis and be provided to students who partake in one of the following experiences

- Embedding themselves in a computational science research group outside their major, or
- Joining a large-scale computational science research project at Argonne, Fermilab, or in a local company.

The former opportunity moves our students outside their silos. The broadening experience will make them more valuable members of interdisciplinary teams performing large-scale computations. It will help them learn and practice Skills 13–16. For example, a natural science student embedded in a mathematical science research group might experiment with how well the group’s new algorithms solve a practical science problem. A computer science student embedded in a natural science research group might lead the adaption of a scientific application from a single core to a high-performance computing architecture.

CISC began a series of lunchtime matchmaking seminars in the fall of 2017 to introduce computational scientists and engineers at Illinois Tech to each others’ research. CISC also hosted a seed grant competition and awarded one grant to an interdisciplinary team. These continuing activities will generate more opportunities for CISC fellows and more quality fellowship applications. Moreover, the CISC fellowships, along with the matchmaking seminars and the seed grants, will help CISC achieve its goal to promote more competitive proposals for external funding of interdisciplinary computational science research.

The ongoing large-scale computational projects at Argonne and Fermilab will provide students the opportunity to learn and practice Skills 4, 5, 6, 14, and 15. Argonne has several software development projects including PETSc [2] and xSDK [4]. Fermilab has a heterogeneous computing environment for high energy physics (HEPCloud) [10] under continuous development. Two of our external advisory board members, Lois Curfman McInnes (Argonne) and Burt Holzman (Fermilab) will aid us in placing CISC Fellows in these two national labs.

The (co-)PIs have had collaborations and contacts with Chicago-area companies involved in supporting or performing large-scale calculations as part of their research, such as HAVI (see Sect. 3.4.1) and NAG. We will further develop these contacts and identify sources of possible projects for CISC fellows.

At the end of the summer the CISC fellows will present their work in posters, possibly coinciding with the presentations by the COD students. During their presentations the CISC Fellows will interact with the summer computational science course students (Sect. 3.3).

Each year, starting in 2019, we will offer CISC fellowships to two or more undergraduates (\$5000 stipend each) and five or more graduate students (\$7000 stipend each). Announcement of the fellowships will be made to science students near the beginning of the calendar year. The students applying will provide a synopsis of their proposed projects, their curriculum vitae, and letters of recommendation from their advisor and the project supervisor. The (co-)PIs will decide who will receive the fellowships. Brochures describing the CISC fellowships will be provided to the departments to use in recruiting new students with interest in computational science.

By the time that this grant is completed, we expect that our overall training program for computational scientists will have prepared a significant number of our students to compete for funded internship opportunities in the national laboratories. Moreover, we anticipate that embedding students in research groups outside their silos will spark interdisciplinary collaborations among research groups. This in turn will lead to external funding of interdisciplinary computational projects that will draw together and financially support computational science students from diverse majors.

4. BROADER IMPACTS

4.1. Educating Next-Generation Computational Scientists. Standard science curricula have not kept pace with the potential of advanced CI for computational science. Computer science courses teach the architectures and languages for large-scale computation, but they do not teach their students how to team up with domain experts to solve important scientific problems. Computational mathematics, biology, chemistry, and physics courses do teach computing, but not educate their students enough in advanced computing technologies and computation thinkings to lead the use and contribution to advanced CI.

Our proposed computational science education program will set students on a path so that by the time they finish their PhD studies, they can be cross-disciplinary computational scientists making vital contributions to interdisciplinary research teams discovering new science. Their work will lead to breakthroughs in our understanding nature, from the sub-atomic to the astrophysical scale. Their work will lead to breakthroughs in health, medicine, manufacturing and finance, which will improve the quality of our lives and the national economy.

The students leaving our programs will carry their academic and practical knowledge of advanced CI to their new research groups in universities, government labs, and companies that they will join. They will share what they have learned with those who need to use more powerful computing capability, but do not yet know how. The knowledge that we impart will be multiplied.

4.2. Impacting Research in Computer Science, Mathematical Science, and Natural Science. The interdisciplinary computational research promoted by our program will generate interesting research problems for the contributing disciplines:

- Complex quantitative models of natural science phenomena will challenge computational mathematicians and computer scientists to develop new algorithms that can utilize more advanced software and hardware.
- The advances in CI will alter how computational mathematicians and computer scientists design algorithms and measure algorithmic efficiency. Efficiency measures for algorithms running on single processors do not apply to advanced CI, and new efficiency measures are needed. Given the variety of architectures and languages, operation count or clock time are not adequate good choices.
- Large-scale scientific computation enabled by advanced CI will answer existing questions in natural science and prompt new research questions in these sciences.

The computational scientists educated by our program will be prepared to tackle these new research problems because they will be proficient contributors and users of advanced CI, i.e., CICs and CIUs. They will know their own disciplines deeply enough to have the new ideas required to solve these new problems. Because they are cross-disciplinary, they will understand how their expertise meshes with the ideas from other disciplines to solve the problem at hand.

Consider an example involving quasi-Monte Carlo (qMC) methods (Hickernell’s research), which are used to evaluate multi-dimensional integrals. From their 1960s until the early 1990s the progress in qMC was mainly theoretical. Then in the mid-1990s, Paskov and Traub [38] showed that qMC could price a collateralized mortgage obligation much more efficiently than independent and identically distributed Monte Carlo (IIDMC). This was surprising, since existing theory supporting showed that qMC was better than IIDMC up to dimension only five or so, whereas Paskov and Traub’s problem had dimension 360. This computational result challenged mathematicians and theoretical computer scientists to re-visit the long-held theory, and the result was dozens of articles and a more complete understanding of the situations in which qMC is superior to MC, even for infinite dimensional integration. New computation raised new research questions, which were then answered.

As another example, a couple of years ago, Jiménez Rugama, the former PhD student of Hickernell, was invited to join a research group of physicists at Fermilab to explore whether their Monte Carlo (MC) computations could be sped up. The default IIDMC algorithm being used was several decades old. Jiménez Rugama was able to introduce the Fermilab group to the much more efficient qMC methods. In the course of implementing these methods, Jiménez Rugama observed an anomaly in the calculations, which was traced to a bug in the long-standing physics code that generated the particle collision events.

4.3. Modeling How Silos Can be Broken. We recognize that computer science, mathematical science, and the several natural sciences are well-defined disciplines. Someone living in one of these silos cannot take full advantage of advanced CI for scientific computation. The solution is not to build a new computational science silo. The solution is to develop cross-disciplinary computational scientists that are deep in one discipline but break out of their own silo to be conversant in others.

This project will show other educational institutions how to break down the major discipline silos to foster high quality computational science. The activities that we will implement will be models for others. We will make our syllabi, training materials, and procedures available on repositories for others to follow and model. (See the Data Management plan for more specifics.) We will publicize our successes and the lessons learned from failures in conference talks. We will advise those interested to adopt our best practices.

4.4. Codifying Good Computational Science Practice. The practice of good computational science is built upon diverse ideas that are not necessarily contained in one text or reference book. Some texts on computational science, such as [47], focus on numerical methods, while others, such as [46], focus on scientific applications. Neither of these types of texts give significant attention to parallel computing or software engineering for high performance computational science software libraries. Texts on high performance computing for computational scientists, such as [30], are incomplete in their coverage of numerical methods or important applications.

As we initiate, develop, and expand the new courses described in Sects. 3.4.3–3.4.5 and as we work with the CISC fellows, we will gather digestible resources describing good computational science practice. These resources will be referenced on the CISC webpage and published in journal articles.

4.5. Broadening Access to Computational Science Opportunities. Illinois Tech’s summer programs are aimed at serving the Chicago area high school students, especially those who are underrepresented minorities. The goal is to provide intellectually stimulating experiences for them that will better prepare them for undergraduate and graduate study. In particular, our summer course in computational science (see Sect. 3.3) will broaden access to a career in computational science. By teaching our students how compute answers to scientific problems *and* giving them two college credits as well, we will be giving our students a head start that they would not have otherwise.

Community colleges enroll approximately 45% of the nation’s undergraduate students, a disproportionate number of which are underrepresented minorities [26, 37]. Nearly 90% of these students have a goal of transferring to four-year institutions to complete their bachelor’s degree, however the actual transfer rate is estimated to be only 25–40% [21, 32]. Studies have shown that involving students, especially those from underrepresented groups, in research activities decreases their attrition rate and increases the probability they will pursue further education [3, 25]. Our research experience for CoD students (Sect. 3.5) will ignite their interest in computational science, giving them motivation to complete their undergraduate degrees and pursue advanced degrees.

4.6. Sharing What We Have Learned. In the course of carrying out our proposed work, we will be learning from

- The background work we do in preparing our new and enhanced courses,

- The ongoing formal assessment that we will undertake (see Sect. 6),
- The continual advice from our external advisors (see Sect. 5),
- Informal student feedback, and
- Our experience.

We will distill the insights gained and share them in articles, talks, and posts on our website. We will seek out venues where our experience can gain a broader audience, such as conferences, websites, and online forums aimed at (potential) computational science educators. We will also make available the course materials, software packages, and code templates that we have developed so that they are available to others.

5. PARTNERSHIPS AND EXTERNAL ADVISORY BOARD

To maximize the impact of our project, we are partnering with Ms. April Welch, Associate Vice-President of Strategic Initiatives at Illinois Tech. Ms. Welch and her colleagues in Illinois Tech's Office of Enrollment will assist us in recruiting students for the Summer Computational Science Course (see Sect. 3.3).

Ms. Welch and her colleagues have established multiple programs for recruiting and supporting high school students with disadvantaged socio-economic backgrounds and preparing them to succeed at a rigorous university like Illinois Tech. The Undergraduate Admissions Office has a network of staff members and alumni actively recruiting African-American and Hispanic students for admission to Illinois Tech and the summer programs that her office supports.

We are also partnering with three other organizations through key contacts:

- Dr. Tom Carter, Professor of Physics at College of DuPage (CoD), the largest community college in Illinois, located in the western suburbs of Chicago,
- Dr. Lois Curfman McInnes, Senior Computational Scientist in the Mathematics and Computer Science Division at Argonne National Laboratory, where she was former PETSc co-lead and is presently xSDK co-lead, and
- Dr. Burt Holzman, Assistant Director of the Scientific Computing Division at Fermi National Accelerator Laboratory (Fermilab), where he oversees the HEPcloud program and coordinates cross-cutting initiatives and solutions across the facility.

Dr. Carter will partner with us to identify CoD students most suited to take advantage of our summer research experience for under-served undergraduates (see Sect. 3.5). He will help communicate to the students our expectations for the summer and feedback to us any difficulties that the students are having. He will also let us know these students' next steps after finishing their CoD studies.

Dr. Curfman McInnes will help us identify large-scale computation projects and advisors at Argonne that are suitable for CISC Fellows (see Sect. 3.6). Particular attention will be given to PETSc and xSDK related projects. She will advise in the selection of the CISC Fellows to be placed at Argonne.

Dr. Holzman will help us identify large-scale computation projects and advisors at Fermilab that are suitable for CISC Fellows (see Sect. 3.6). Particular attention will be given to HEPcloud projects involving heterogeneous systems. He will advise in the selection of the CISC Fellows to be placed at Fermilab.

Our External Advisory Board will be comprised of Dr. Carter, Dr. Curfman McInnes, and Dr. Holzman. This board will meet with the (co-)PIs and Senior Personnel twice a year (by teleconference if necessary), to review the progress of our project and advise us on how our efforts can be even more effective. Apart from these formal meetings, we will welcome their advice at any time and solicit their advice as needed.

6. ASSESSMENT

The proposed project is both ambitious and complex. There are four levels of participants (i.e., high school students, community college students, undergraduate students, and graduate students) as well as participants that are classified as cyberinfrastructure (CI) users (CIUs) and contributors (CICs). There are 16 different learning outcomes, all of which are performance tasks as opposed to understandings that are assessed with paper and pencil exams. In addition, the proposed project will also track the post project trajectory of the various participant levels (e.g., future course enrollments, employment situations, etc.). In order to insure the trustworthiness of data collected, all assessments will have established content validity as well as inter-rater agreement (i.e., reliability).

6.1. Content Validity of Assessments. The specific learning outcomes (see Sect. 3.1) are all performance tasks. Some are independent of each other and some are interrelated. Consequently, a single assessment instrument would not be appropriate. Rather, assessment tasks will need to be developed for each performance task and validity will need to be established for each task. A group of five individuals, derived from the project staff, will develop a task purported to assess each of the learning objectives. After development, Advisory Board members (and additional experts in the field if needed) will independently assess whether each item assesses what it purports to measure. An agreement level of 80% will be desired. Five individuals, as is convention, will be needed for this assessment. If an agreement level of 80% is not reached, the task will be revised and subjected to another independent evaluation. This process will be repeated until an agreement level of 80% is reached for each performance task.

6.2. Inter-Rater Agreement (Reliability) of Assessments. It is one thing to establish whether an assessment task is aligned with the learning outcome for which it is designed, but it is another to document that the assessment of student/participant performance is consistent. Given that all of the learning outcomes are performance tasks, the consistency of assessment will need to be established by documenting that those individuals evaluating the task performance do so in a consistent manner. A scoring rubric will need to be developed for each performance task. This will involve the specification of what each respondent will need to include in their performance of the specified task. In short, the rubric defines what should be included for a particular “grade” on the task. Sample student responses (e.g., 3–5) will be evaluated by a group of scorers (preferably five) to document whether there is agreement in how a student’s performance is evaluated. By convention, the target level of agreement should be at least 80% for each performance task. This will establish that students’ performance on the learning tasks are being scored in a consistent manner. Once inter-rater agreement is established it is not necessary for every student task to be scored by multiple evaluators. Ordinarily, inter-rater agreement is established at the beginning of an investigation. However, in this case, every effort will be made to check this consistency midway through the scoring of student tasks and at the end of the evaluation process.

It is important to note that high school students, community college students, undergraduate students, and graduate students are not expected to achieve each of the learning outcomes at the same level. Consequently, the sample student responses scored for reliability will be performed separately for each student level.

6.3. Research Questions and Data Analysis. The sample for the proposed project consists of four different levels of students. The accomplishments of these four groups of students will be analyzed independently. That is, their performance levels for each of the 16 learning outcomes will be determined independently. Clearly, students at these different levels cannot be expected to perform at the same level on each of the learning outcomes. Additionally, it would be inappropriate to assume that contributors (CICs) and users (CIUs) would be expected to perform at the same level.

Although it would not be likely that the students would have the ability to perform the learning outcomes in advance of the program, using a pretest/posttest design will potentially provide important information relative to the range of students involved in the project. changes will not be determined. Since categorical data (students with varying abilities to complete the performance tasks) simple percentages of performance will be documented to establish success of the proposed project. Specially, Chi-Square tests will be performed comparing the pretest scores with posttest scores with a significance level of 0.05 considered to be statistically significant.

Project participants will be located in different contexts (e.g., research labs, IIT, and community colleges, high schools), hence it will be of interest to see if there are differences in performance relative to such contexts. Consequently, Chi-Square comparisons ($p < 0.05$) will be used to compare differences in learning outcome performances on each of the learning outcomes at each of the different educational levels. Such comparisons may have implications for future interventions for students at various academic levels.

Finally, it will be of particular interest to follow the professional trajectories of participants in this proposed program following their participation. For example, do high school students enter university programs related to CI following their participation or do they become directly employed in the field through various occupations? In a similar manner it will be of interest to follow the trajectories of community college, undergraduate, and graduate students. What, if any, was the impact of participation in the program? Data on post program trajectories will be compiled with simple frequencies related to university and/or career paths. Given that the proposed program is a three year program, participants will be followed for a period of up to two years after participation.

One of the most intriguing driving motivations for the proposed project is the breaking down of disciplinary silos with respect to the program focus on an area that is clearly interdisciplinary. Participants in this project will come from a variety of subject matter areas, as well as different educational levels. It is unclear if there is any relationship among disciplinary areas, educational levels, and potential professional trajectories within CI. One final focus of data analysis, although clearly exploratory, will be a correlation matrix that includes educational level, disciplinary background, program success (with respect to outcomes), and career trajectory. Depending on the results of such an analysis, further more targeted analyses may be pursued.

6.4. Tracking the Career Outcomes of Our Students. In addition to assessing the skills learned by our students, we plan to track their career paths. We will survey our students to identify their next program of studies or their job, whichever the case may be. This will allow us to determine how many pursue careers as computational scientists. We will also ask them what skills they are finding important in their next degree program or job.

7. TIMELINE

The chart below highlights when certain project tasks are expected to be performed.

Time	Sect.	Task
2018		
November		Proposed work begins
	5	Meet with CISC External Advisory Board—confirm priorities and first steps
	3.3	Prepare advertising for the summer computational science course
	3.5	Prepare advertising for the research experience for CoD students
	3.6	Advertise the CISC summer fellowships
	3.4.1	Post information on existing computational science courses on the CISC website

2019		
January	3.3	Begin promoting summer computational science course
	3.4.3	Pilot new professional practices for computational science course
	3.4.4	Offer undergraduate parallel/distributed computing course
	3.5	Begin promoting research experience for CoD students
February	3.6	Identify potential projects/research groups to receive CISC fellows
	3.6	Accept applications for CISC summer fellowships
	3.4.1	Begin seeking more industrial partners to propose course progress
March	3.6	Announce CISC summer fellowship awardees
	3.5	Announce COD students accepted for the summer research experience
May	3.5	Begin research experience for CoD students; start with the “crash” course
	3.6	Begin summer CISC fellowships
July	3.3	Begin summer computational science course
August	3.4.5	Offer advanced scientific computing course
September	5	Meet with CISC External Advisory Board
	3.3	Evaluate summer computational science course
	3.5	Evaluate research experience for CoD students
	3.6	Evaluate summer CISC fellowships
October	4.6	Ensure that materials developed during the first year are published on the web
December	3.3	Begin promoting summer computational science course
	3.5	Begin promoting research experience for CoD students
	3.6	Identify potential projects/research groups to receive CISC fellows
<hr/>		
2020		
January		Same schedule for major course offerings and summer activities as in 2019,
– 2021		but with promotion for the summer activities starting the December be-
October		forehand
	4.6	Analyze assessment data and prepare publications documenting what has been learned
	4.6	Attend conferences to share what has been learned
<hr/>		