New Results on Complex Golay Pairs

Curtis Bright, Vijay Ganesh, Albert Heinle and Ilias Kotsireas

Abstract. We verify that 23 is not a complex Golay number, or equivalently that complex Golay sequences of order 23 do not exist. This confirms the conjecture of Craigen, Holzmann, and Kharaghani [4] from 2002. Additionally, we present a new algorithm for exhaustively searching for complex Golay sequences of a given order and provide for the first time an enumeration of all complex Golay sequences of orders 20, 22, and 24.

Keywords. Complex Golay sequences, Exhaustive search, Autocorrelation, Diophantine systems.

1. Introduction

Complex Golay sequences have been extensively studied in [6], [3], [4]. They were introduced in order to expand the orders of Hadamard matrices attainable via (ordinary) Golay sequences, also called Golay pairs. A notion of canonical form for Golay sequences has been introduced in [11] and representatives of the equivalence classes of Golay sequences for all lengths ≤ 40 have been found. Golay and Turyn have shown how to multiply Golay sequences of length g_1 with Golay sequences of length g_2 in order to construct Golay sequences of length $g_1 g_2$. Golay sequences, have been classified up to order 100 in [2] where the authors show that all such pairs can be derived using certain equivalence and composition operations from five primitive Golay pairs.

A positive integer g is called a complex Golay number, if there exist complex Golay sequences of order g. A multiplication for complex Golay sequences, similar to the one for Golay sequences mentioned above, cannot exist, since $3, 5$ are complex Golay numbers, but 15 is not.

The fundamental paper [4] contains exhaustive searches for all lengths of complex Golay sequences up to 19, a partial search for orders 20 and 22 and states that the authors suspect they do not exist for length 23. In addition, it is shown that $g = 7, 9, 14, 15, 17, 19, 21$ are not complex Golay numbers. Based on the numerical evidence they have gathered, the authors state four conjectures pertaining to complex Golay sequences and complex Golay numbers, all of which are still open. The fourth of their conjectures states that "every prime divisor of a complex Golay number is a complex Golay number". Given the aforementioned list of known complex Golay numbers, this means that to disprove this conjecture, one would have to construct/find complex Golay sequences of any of the orders 28, 34, 35, 38, 46, $\ldots$. Another interesting phenomenon is that the only known odd complex Golay numbers are all primes. Therefore it would be of interest to know whether 25, 27, 33, 35, 39 are complex Golay numbers or not. Finally, the authors provide theorems on the algebraic structure of complex Golay sequences which connects their structure to polynomial factorization over finite fields. In certain cases these theorems can be used to speed up computational algorithms to search for these rather elusive combinatorial objects, although the present work does not require them.

The following theorem [4, Cor. 13] shows the importance of complex Golay numbers to Hadamard matrices, which are combinatorial objects constructed [1] for their use in error correcting codes and studied [5] for their many elegant mathematical uses and properties.

Theorem 1.1. *If g is a complex Golay number, then there exists a Hadamard matrix of order $4g$.*

In particular, this holds for $g = 2^{a+u}3^b 5^c 11^d 13^e$ where the variables in the exponents are non-negative integers which satisfy some linear inequalities. In view of Theorem 1.1 it becomes apparent that extending the list of known prime complex Golay numbers would entail strengthening this theorem by enlarging the set of attainable orders of Hadamard matrices constructible via complex Golay sequences.

We present our new algorithm for exhaustively searching for complex Golay sequences of a given order in Section 3, following the necessary background covered in Section 2. We show that by solving certain Diophantine systems one can derive restrictions on the possible forms that all complex Golay sequences of a given order must satisfy. These restrictions are then used along with a procedure which can generate all permutations of a given form; this allows an exhaustive search to be performed on a space smaller than would be necessary using a naive exhaustive search.

2. Background on Complex Golay Sequences

In this section we present the background necessary to describe our algorithm for enumerating complex Golay sequences. First, we require some preliminary definitions to describe the kind of sequences we will be searching for.

Definition 2.1 (cf. [9]**).** The *complex aperiodic* (or *complex nonperiodic*) autocorrelation function of a sequence $A = [a_1, \dots, a_n] \in \mathbb{C}^n$ of length $n \in \mathbb{N}$ is defined as

$$N_A(s) := \sum_{k=1}^{n-s} a_k \overline{a_{k+s}}, \qquad s = 0, \dots, n-1.$$

Definition 2.2 (cf. [9]**).** Two sequences A and B in $\mathbb{C}^n$ are said to have *constant aperiodic autocorrelation* if there is a constant $c \in \mathbb{C}$ such that

$$N_A(s) + N_B(s) = c, \qquad s = 1, \dots, n-1.$$

Definition 2.3. A pair of sequences (A, B) with A and B in $\{\pm 1, \pm i\}^n$ are called a *complex Golay sequence pair* if they have zero constant aperiodic autocorrelation, i.e.,

$$N_A(s) + N_B(s) = 0, \qquad s = 1, \dots, n-1.$$

If such sequences exist for $n \in \mathbb{N}$ we call n a *complex Golay number*.

Note that if A and B are in $\{\pm 1, \pm i\}^n$ then $N_A(0) + N_B(0) = 2n$ by the definition of the complex aperiodic autocorrelation function and the fact that $x\overline{x} = 1$ if x is ± 1 or $\pm i$.

2.1. Equivalence Operations

There are certain invertible operations which preserve the property of being a complex Golay sequence pair when those operations are applied to sequence pairs (A, B). These are summarized in the following proposition.

Proposition 2.4 (cf. [4]**, section 4).** *Let $([a_1, \dots, a_n], [b_1, \dots, b_n])$ be a complex Golay sequence pair. The following are then also complex Golay sequence pairs:*

E1. *(Reversal)* $([a_n, \dots, a_1], [b_n, \dots, b_1])$.
E2. *(Conjugate Reverse A)* $([\overline{a_n}, \dots, \overline{a_1}], [b_1, \dots, b_n])$.
E3. *(Swap)* $([b_1, \dots, b_n], [a_1, \dots, a_n])$.
E4. *(Scale A)* $([ia_1, \dots, ia_n], [b_1, \dots, b_n])$.

E5. *(Positional Scaling)* $([p_1a_1,\dots,p_na_n],[p_1b_1,\dots,p_nb_n])$ *where* $p_k := i^k$.

Definition 2.5. We call two complex Golay sequence pairs (A,B) and (A',B') *equivalent* if (A',B') can be obtained from (A,B) using the transformations described in Proposition 2.4.

2.2. Useful Properties and Lemmas

In this subsection we prove some useful properties that complex Golay sequences must satisfy and which will be exploited by our algorithm for enumerating complex Golay sequences.

The first lemma provides a relationship that all complex Golay sequences must satisfy. To state it, we require the following definition.

Definition 2.6 (cf. [4]**).** The *Hall polynomial* of the sequence $A := [a_1,\dots,a_n]$ is defined to be $h_A(z) := a_1 + a_2z + \cdots + a_nz^{n-1} \in \mathbb{C}[z]$.

Lemma 2.7. *Let* (A,B) *be a complex Golay sequence pair. For every* $z\in\mathbb{C}$ *with* $|z| = 1$*, we have*

$$|h_A(z)|^2 + |h_B(z)|^2 = 2n.$$

Proof. Since $|z| = 1$ we can write $z = e^{i\theta}$ for some $0 \le \theta < 2\pi$. Similar to the fact pointed out in [7], using Euler's identity one can derive the following expansion:

$$|h_A(z)|^2 = N_A(0) + 2\sum_{j=1}^{n-1}\big(\mathrm{Re}(N_A(j))\cos(\theta j) + \mathrm{Im}(N_A(j))\sin(\theta j)\big).$$

Since A and B form a complex Golay pair, by definition one has that $\mathrm{Re}(N_A(j)+N_B(j)) = 0$ and $\mathrm{Im}(N_A(j)+N_B(j)) = 0$ and then

$$|h_A(z)|^2 + |h_B(z)|^2 = N_A(0) + N_B(0) = 2n. \qquad \square$$

This lemma is highly useful as a condition for filtering sequences which could not possibly be part of a complex Golay sequence pair, as explained in the following corollary.

Corollary 2.8. *Let* $A \in \{\pm1,\pm i\}^n$*,* $z\in\mathbb{C}$ *with* $|z| = 1$*, and* $|h_A(z)|^2 > 2n$*. Then* A *is not a member of a complex Golay sequence pair.*

Proof. Suppose the sequence A was a member of a complex Golay sequence pair whose other member was the sequence B. Since $|h_B(z)|^2 \ge 0$, we must have $|h_A(z)|^2 + |h_B(z)|^2 > 2n$, in contradiction to Lemma 2.7. $\square$

The next lemma is useful to derive conditions on how often each type of entry (i.e., 1, -1, i, $-i$) occurs in a complex Golay sequence pair. It is stated in [3] using a different notation; we use the notation $\mathrm{resum}(A)$ and $\mathrm{imsum}(A)$ to represent the real and imaginary parts of the sum of the entries of A. For example, if $A := [1,i,-i,i]$ then $\mathrm{resum}(A) = \mathrm{imsum}(A) = 1$.

Lemma 2.9. *Let* (A,B) *be a complex Golay sequence pair. Then*

$$\mathrm{resum}(A)^2 + \mathrm{imsum}(A)^2 + \mathrm{resum}(B)^2 + \mathrm{imsum}(B)^2 = 2n.$$

Proof. Using Lemma 2.7 with $z = 1$ we have

$$|\mathrm{resum}(A) + \mathrm{imsum}(A)i|^2 + |\mathrm{resum}(B) + \mathrm{imsum}(B)i|^2 = 2n.$$

Since $|\mathrm{resum}(X) + \mathrm{imsum}(X)i|^2 = \mathrm{resum}(X)^2 + \mathrm{imsum}(X)^2$ the result follows. $\square$

The next lemma provides some normalization conditions which can be used when searching for complex Golay sequences up to equivalence; we say that a complex Golay sequence is *normalized* if it meets these conditions.

Lemma 2.10. *Let (A,B) be a complex Golay sequence pair. Then (A,B) is equivalent to a complex Golay sequence pair (A',B') which satisfies the conditions*

$$0 \le \operatorname{resum}(A') \le \operatorname{imsum}(A'),$$
$$0 \le \operatorname{resum}(B') \le \operatorname{imsum}(B'),$$
$$\textit{and } \operatorname{resum}(A') \le \operatorname{resum}(B').$$

Proof. We will transform a given complex Golay sequence pair (A,B) into an equivalent normalized one using the equivalence operations of Proposition 2.4. To start with, let $A' := A$ and $B' := B$.

First, we ensure that $|\operatorname{resum}(A')| \le |\operatorname{imsum}(A')|$. If this is not already the case then we apply operation E4 (which has the effect of switching $|\operatorname{resum}(A')|$ and $|\operatorname{imsum}(A')|$) and the updated A' will satisfy this condition.

Next, we ensure that $\operatorname{resum}(A') \ge 0$. If this is not already the case then we apply operation E4 twice (which has the effect of negating each element of A') and the updated A' will satisfy $0 \le \operatorname{resum}(A') \le |\operatorname{imsum}(A')|$. If $\operatorname{imsum}(A') \ge 0$ then the first condition is satisfied. If not, then it will be satisfied after applying operation E2 (which negates $\operatorname{imsum}(A')$).

Next, we ensure that the second condition holds. If it not already the case, then we apply operation E3 (switch A' and B'); this will cause the second condition to be satisfied at the cost of causing the first condition to no longer be satisfied. However, we may now repeat the above directions to make the first condition satisfied again; note that these directions do not modify B' so that once we have completed them both the first two conditions will be satisfied.

Lastly, we ensure the final condition $\operatorname{resum}(A') \le \operatorname{resum}(B')$. If it is not already satisfied then we apply E3 (switch A' and B') and the updated sequence pair will satisfy the condition as required. □

2.3. Sum-of-square Decomposition Types

A consequence of Lemma 2.9 is that every complex Golay sequence yields a decomposition of $2n$ into a sum of four squares. With the help of a computer algebra system (CAS) one can even enumerate all the ways that $2n$ may be written as a sum of four squares. Furthermore, since it suffices to search for complex Golay sequence pairs up to equivalence, by Lemma 2.10 we can make assumptions about the form of the decomposition, for example, that the resum and imsum of A and B are non-negative. Thus, it suffices to use a CAS to solve the quadratic Diophantine system

$$r_a^2 + i_a^2 + r_b^2 + i_b^2 = 2n, \qquad 0 \le r_a \le i_a, \qquad 0 \le r_b \le i_b, \qquad r_a \le r_b \tag{2.1}$$

for indeterminants r_a, i_a, r_b, $i_b \in \mathbb{Z}$.

Example 1. When $n = 23$ the Diophantine system (2.1) has exactly four solutions, as given in the following table:

r_a	i_a	r_b	i_b
0	1	3	6
1	2	4	5
0	3	1	6
1	4	2	5

Let (A,B) be a complex Golay sequence of order n. Furthermore, let u, v, x, and y represent the number of 1s, -1s, is, and $-i$s in A, and let r_a and i_a represent the resum and imsum of A, respectively. We have that

$$u, v, x, y \ge 0, \qquad u - v = r_a, \qquad x - y = i_a, \qquad u + v + x + y = n. \tag{2.2}$$

Given the values of n, r_a, and i_a this is a system of linear Diophantine equations which is to be solved over the non-negative integers. From the last equality we know that $u, v, x, y \le n$ so such a system necessarily has a finite number of solutions.

Example 2. When $n = 23$, $r_a = 0$, and $i_a = 1$, the Diophantine system (2.2) has exactly 12 solutions, as given in the following table:

u	v	x	y
0	0	12	11
1	1	11	10
2	2	10	9
3	3	9	8
4	4	8	7
5	5	7	6
6	6	6	5
7	7	5	4
8	8	4	3
9	9	3	2
10	10	2	1
11	11	1	0

The *multinomial coefficient* $\binom{n}{u,v,x,y} = \frac{n!}{u!v!x!y!}$ tells us how many possibilities there are for $A \in \{\pm 1, \pm i\}^n$ with u entries which are 1s, v entries which are -1s, x entries which are is, and y entries which are $-i$s. For example, there are $\frac{23!}{12!11!} = 1{,}352{,}078$ possibilities for A with 12 entries which are is and 13 entries which are $-i$s (i.e., those which correspond to the first row of the table in Example 2). Algorithms for explicitly generating all such possibilities for A can be found in [8].

3. Description of Our Algorithm

First, we fix an order n for which we are interested in generating a list of inequivalent complex Golay sequence pairs (A, B). Our algorithm finds all solutions r_a, i_a, r_b, i_b of (2.1) and for all pairs (r_a, i_a) then solves the system (2.2). For each solution quadruple (u, v, x, y) we use Algorithm 7.2.1.2L from [8] to generate all possibilities for A with the appropriate number of 1s, -1s, is, and $-i$s. For each possibility for A we compute $H_k := |h_A(e^{2\pi ik/50})|^2$ for $k = 1, \ldots, 49$. If any value of H_k is strictly larger than $2n$, we immediately discard the sequence A (see Corollary 2.8); if all values of H_k are smaller than $2n$ then we record the sequence A as one which could appear in the first position of a complex Golay sequence pair.

Next, we repeat the above steps except that we solve the system (2.2) for all pairs (r_b, i_b) (replacing r_a with r_b and i_a with i_b), and this time we generate a list of possibilities for B, sequences which could appear in the second position of a complex Golay sequence pair.

Finally, we use the matching technique of [10] to compile a list of all complex Golay sequence pairs of a given order. We form the strings

$$\mathrm{Re}(N_A(1)), \mathrm{Im}(N_A(1)), \ldots, \mathrm{Re}(N_A(n-1)), \mathrm{Im}(N_A(n-1))$$

and

$$-\mathrm{Re}(N_B(1)), -\mathrm{Im}(N_B(1)), \ldots, -\mathrm{Re}(N_B(n-1)), -\mathrm{Im}(N_B(n-1))$$

for all possibilities for A and B which were previously generated. We then create two files, those containing the 'A' strings sorted in lexicographic order, and those containing the 'B' strings sorted in lexicographic order. Finally, we perform a linear scan through the files to find all the strings which are common to both files. All matches are guaranteed to produce complex Golay sequences since if the strings derived from sequences A and B matched then $\mathrm{Re}(N_A(s)) + \mathrm{Re}(N_A(s)) = \mathrm{Im}(N_A(s)) + \mathrm{Im}(N_A(s)) = 0$ for $s = 1, \ldots, n-1$. Furthermore, all normalized complex Golay sequences will be among the matches since by construction if (A, B) is a normalized complex Golay sequence then A appears in the first list of possibilities generated and B appears in the second list of possibilities which were generated.

If one wants the complete list of complex Golay sequence pairs, one can now repeatedly apply the equivalence operations E1–E5 to the list of normalized complex Golay sequence pairs until those operations no longer produce new sequences. Note that normalized complex Golay sequences are not necessarily inequivalent, as the normalization conditions of Lemma 2.10 do not capture all possible equivalences. If there are two sequences which are equivalent to each other this can be checked by applications of the equivalence operations E1–E5 and one of the equivalent sequences can be removed if desired.

3.1. Optimizations

There are some further properties which while not essential to the algorithm can be exploited to remove some extraneous computations.

Lemma 3.1. *Let $H_k := |h_A(e^{2\pi ik/50})|^2$ be the quantity which we use in our algorithm's filtering criterion, and let H'_k be the same quantity but computed with respect to A', the reverse of A. Then $H_k = H'_{50-k}$.*

Proof. Let $\theta := 2\pi k/50$. Then $H'_{50-k} = |h_{A'}(e^{-i\theta})|^2$ and as in the decomposition in Lemma 2.7 one has

$$|h_{A'}(e^{-i\theta})|^2 = N_{A'}(0) + 2\sum_{j=1}^{n-1}\big(\mathrm{Re}(N_{A'}(j))\cos(-\theta j) + \mathrm{Im}(N_{A'}(j))\sin(-\theta j)\big).$$

From the definition of the aperiodic autocorrelation function one sees that $N_{A'}(0) = N_A(0)$ and $N_{A'}(s) = \overline{N_A(s)}$ for $s = 1, \ldots, n-1$. Using this with the standard facts that $\cos(-x) = \cos(x)$ and $\sin(-x) = -\sin(x)$ one derives that this expansion is exactly the same as the expansion for $|h_A(e^{i\theta})|^2 = H_k$, as required. □

In light of Lemma 3.1, we do not need to compute the values H_k for both A and its reverse, since the H_k values for the reverse of A will be the exact same as those for A (albeit in reverse order). In other words, A will be discarded by our filtering condition if and only if its reverse is discarded by our filtering condition, so once A has been checked we need not also check its reverse.

To avoid extraneous computations, we only perform the filtering check on one of A and the reverse of A, whichever is lexicographically greater (if A is equal to its reverse it is also checked). Once the filtering process has been completed we take the list of sequences which passed the filter and add to the list the reverse of each sequence on the list (except for those which are their own reverse).

Of course, we can perform the same optimization when performing the filtering check on the B sequences as well. In this case one can completely discard sequences whose reverses are lexicographically strictly smaller than themselves because of the following lemma.

Lemma 3.2. *The following normalization condition may be added to Lemma* 2.10*:*

$$B' \geq_{\text{lex}} \text{reverse}(B'). \tag{3.1}$$

Proof. Continuing the proof of Lemma 2.10, if the complex Golay sequence pair (A', B') satisfies (3.1) then we are done. If not, we apply equivalence operation E1 (reversal) to (A', B') so that (3.1) is satisfied. Furthermore, all the normalization conditions of Lemma 2.10 remain satisfied because the operation E1 does not change the resum or imsum of A' or B'. □

Finally, we note that it is possible to optimize the evaluation of the Hall polynomial by reusing previously computed values. If $A := [a_1, \ldots, a_n]$ is the sequence which we need to check the filtering condition for, then we want to compute the Hall polynomial evaluation

$$h_A(e^{2\pi ik/50}) = \sum_{j=0}^{n-1} a_{j+1}e^{2\pi ijk/50} \qquad \text{for } k = 1, \ldots, 49.$$

Because of the periodicity $e^{2\pi ijk/50} = e^{2\pi i(jk \bmod 50)/50}$ and the fact $-e^{2\pi ijk/50} = e^{2\pi i(jk+25)/50}$ there are only 100 possible values for the summand in this sum, namely $xe^{2\pi iy/50}$ for $x = 1$ or i and $y = 0, \dots, 49$. These can be computed once at the start of the algorithm and reused as necessary.

Furthermore, in many cases it is possible to reuse some computations from the Hall polynomial evaluation of the previously checked sequence. The algorithm we used to generate the sequences [8, §7.2.1.2] generates them in lexicographically increasing order, meaning that consecutively generated sequences often share a large common prefix. If the entries $a_1, \dots, a_l$ of a sequence are identical to those in the previously generated sequence then the partial sum $\sum_{j=0}^{l-1} a_{j+1} e^{2\pi ijk/50}$ can be reused, assuming it was computed and stored; as the Hall polynomial evaluations are being computed one can remember their partial sums for varying l and k in a table.

4. Results

The algorithm described above was implemented in C and run for all orders n up to 24. Our main new finding is that $n = 23$ is not a complex Golay number, i.e., that complex Golay sequences of order 23 do not exist. This result confirms the conjecture of [4] and in addition it implies that the next candidate prime complex Golay number is $n = 29$. Our results also include a complete search for all orders up to 24; this search had already been completed in [4] for all orders up to 19 as well as 21. Our results match the previously computed results in all cases, but we also provide complete results for orders 20, 22, 23, and 24.[1] The computations were performed with an Intel Xeon CPU running at 3.3GHz under Ubuntu 14.04. The algorithm's run time in hours for orders 20, 21, 22, 23, and 24 was 4, 13, 32, 179, and 361, respectively.

Figure 1 contains a table summarizing how many complex Golay pairs exist for each order up to 24. The second column contains the total number of complex Golay pairs and the third column contains the number of inequivalent complex Golay pairs. We have also performed some preliminary searches for complex Golay sequences of orders 25 to 28. Except for a class of solutions equivalent to a real Golay sequence of order 26 these searches did not yield any complex Golay sequences.

Order	Total Pairs	Inequiv. Pairs
1	16	1
2	64	1
3	128	1
4	512	2
5	512	1
6	2048	3
7	0	0
8	6656	17
9	0	0
10	12,288	20
11	512	1
12	36,864	52

Order	Total Pairs	Inequiv. Pairs
13	512	1
14	0	0
15	0	0
16	106,496	204
17	0	0
18	24,576	24
19	0	0
20	215,040	340
21	0	0
22	8192	12
23	0	0
24	786,432	1056

FIGURE 1. A table summarizing the number of complex Golay pairs which exist in all orders up to 24.

[1]These results may be downloaded from the webpage `https://cs.uwaterloo.ca/~cbright/golay/`.

Acknowledgements

The authors would like to thank Ian Li for his help in verifying our algorithm and its implementation.

References

[1] Leonard Baumert, S. W. Golomb, and Marshall Hall. Discovery of an Hadamard matrix of order 92. *Bull. Amer. Math. Soc.*, 68(3):237–238, 05 1962.

[2] P. B. Borwein and R. A. Ferguson. A complete description of Golay pairs for lengths up to 100. *Math. Comp.*, 73(246):967–985, 2004.

[3] R. Craigen. Complex Golay sequences. *J. Combin. Math. Combin. Comput.*, 15:161–169, 1994.

[4] R. Craigen, W. Holzmann, and H. Kharaghani. Complex Golay sequences: structure and applications. *Discrete Math.*, 252(1-3):73–89, 2002.

[5] Jacques Hadamard. Résolution d'une question relative aux déterminants. *Bull. Sci. Math.*, 17(1):240–246, 1893.

[6] W. H. Holzmann and H. Kharaghani. A computer search for complex Golay sequences. *Australas. J. Combin.*, 10:251–258, 1994.

[7] Hadi Kharaghani and Behruz Tayfeh-Rezaie. A Hadamard matrix of order 428. *Journal of Combinatorial Designs*, 13(6):435–440, 2005.

[8] Donald Ervin Knuth. *The Art of Computer Programming. Vol. 4A, Combinatorial Algorithms, Part 1*. Addison-Wesley, Upple Saddle River (N.J.), London, Paris, 2011.

[9] Ilias S Kotsireas. Algorithms and metaheuristics for combinatorial matrices. In *Handbook of Combinatorial Optimization*, pages 283–309. Springer, 2013.

[10] Ilias S Kotsireas, Christos Koukouvinos, and Jennifer Seberry. Weighing matrices and string sorting. *Annals of Combinatorics*, 13(3):305–313, 2009.

[11] Dragomir Ž. Đoković. Equivalence classes and representatives of Golay sequences. *Discrete Math.*, 189(1-3):79–93, 1998.

Curtis Bright
University of Waterloo
e-mail: cbright@uwaterloo.ca

Vijay Ganesh
University of Waterloo
e-mail: vganesh@uwaterloo.ca

Albert Heinle
University of Waterloo
e-mail: aheinle@uwaterloo.ca

Ilias Kotsireas
Wilfrid Laurier University
e-mail: ikotsire@wlu.ca