Supplementary Information:
Wright-Fisher Exact Solver (WFES)

Ivan Krukov, Bianca de Sanctis, and A.P. Jason de Koning

November 15, 2016

Contents

1 Supplementary Methods **1**

1.1 Finite absorbing Markov chain theory 2

1.2 Rapid solution of restricted linear systems 3

1.3 Parameterization of the Wright-Fisher model 3

1.3.1 Haploid model 4

1.4 Rapid calculation of the transition matrix 4

2 Supplementary Results **5**

2.1 Comparison of exact and approximate results 5

2.2 Effect of truncation 6

2.3 Understanding the performance of WFES 6

2.3.1 Definition of sparsity 7

2.3.2 Space complexity reductions from sparsity 8

2.3.3 Peak RAM usage 8

2.3.4 Runtime of component calculations 8

1 Supplementary Methods

To enable scalable computation with population genetic Markov models we developed several approaches to make the computations feasible, as described in the subsequent sections. First, we calculate the relevant Wright-Fisher transition matrix with a recursive algorithm for rapidly computing whole rows, and a row-parallel implementation (section 1.4). Second, we solve restricted linear systems (section 1.2) using LU decomposition followed by back substitution with routines that exploit sparsity and parallelism. By far the most time-consuming step is the LU decomposition itself. Given the LU decomposition of the relevant sparse matrix, the remaining computations take only seconds even for population sizes around $N_e = 100,000$.

WFES is written in C and is designed to be fast, scalable, and easy to modify. Our implementation exploits routines in the `Intel MKL PARDISO` [Intel, Inc., 2016, Schenk et al., 2000, 2001] library, a state of the art linear solver commonly used for high-performance computing applications. For convenience, our distribution (https://github.com/dekoning-lab/wfes) includes the freely-distributable libraries needed to compile and run the program.

1.1 Finite absorbing Markov chain theory

In this section, we describe several well known results from the theory of absorbing Markov chains without giving explicit proofs. These can generally be found in Kemeny and Snell [1960], alongside examples and further reading.

Given the transition matrix of any Wright-Fisher model, P, having two absorbing states (for extinction and fixation) and an effective population size of N_e, we first re-order the rows and columns without changing the actual entries. Following standard theory [Kemeny and Snell, 1960], this re-ordering groups all of the transient states in their original suborder, followed by all of the absorbing states in their original suborder, so that the matrix is represented as

$$P = \left(\begin{array}{cc} Q & R \\ 0 & I_2 \end{array} \right), \tag{1}$$

where Q is a $(2N_e - 1) \times (2N_e - 1)$ matrix of transient-to-transient state transitions, R is a nonzero $(2N_e - 1) \times 2$ matrix of transient-to-absorbing state transitions, I_2 is the 2×2 identity matrix reflecting absorbing-to-absorbing state transitions, and 0 is an $2 \times (2N_e - 1)$ matrix where every entry is 0, reflecting the absorbing-to-transient state transitions. Then

$$\mathbf{N} = \sum_{k=0}^{\infty} Q^k = (I - Q)^{-1} \tag{2}$$

is called the fundamental matrix of the Markov chain. (Note that the symbol $\mathbf{N}$ is used following convention, and is not related to the population size or N_e.) Each entry $\mathbf{N_{ij}}$ of the fundamental matrix gives the expected number of times the chain is in state j, given that the chain started in transient state i. The variance on this is the (i, j)th entry of

$$\mathbf{N_2} = \mathbf{N}(2\mathbf{N_{dg}} - I) - \mathbf{N_{sq}} \tag{3}$$

where $\mathbf{N_{dg}}$ is a diagonal matrix with the same diagonal as $\mathbf{N}$, and $\mathbf{N_{sq}}$ is the Hadamard product of $\mathbf{N}$ with itself, or entry-wise squared.

Summing the ith row will give the expected time until absorption, given that the chain started in transient state i (and unconditional on any specific absorbing state). Equivalently, we can take the ith entry of the vector

$$\mathbf{t} = \mathbf{N1} \tag{4}$$

In biological terms, this is the expected time until either fixation or extinction occurs. The variance on this is the ith entry of

$$\mathbf{v} = (2\mathbf{N} - I_t)\mathbf{t} - \mathbf{t}_{sq} \tag{5}$$

where $\mathbf{t}_{sq}$ is the Hadamard product of $\mathbf{t}$ with itself.

The probability of absorbing in state j having started in state i is the (i, j)th entry of the matrix

$$B = NR \tag{6}$$

In our case, B will be a $(2N - 1) \times 2$ matrix, where the first and second columns correspond to probabilities of extinction and fixation respectively.

The expected number of times the chain is in state j, conditional on starting in transient state i and absorbing in state k is

$$E_{ik}(j) = \frac{B_{jk}}{B_{ik}} \mathbf{N_{ij}} \tag{7}$$

Then the expected number of steps before absorption, conditional on starting in transient state i and absorbing in state k is

$$\sum_{j=1}^{2\mathbf{N}-1} E_{ik}(j) \tag{8}$$

For completeness, we also give this same result in matrix form. Define D_k as a diagonal matrix with diagonal entries b_{jk}, for a fixed absorbing state k. Then the expected number of steps before absorption having started in transient state i and conditional on absorbing in state k is the ith entry of

$$\tilde{\mathbf{t}} = D_k^{-1}\mathbf{N}D_k\mathbf{1} \tag{9}$$

where $\mathbf{1}$ again is a vector of 1s. The variance on this is the ith entry of

$$\tilde{\mathbf{v}} = (2D_k^{-1}\mathbf{N}D_k - I_t)\tilde{\mathbf{t}} - \tilde{\mathbf{t}}_{sq} \tag{10}$$

Higher moments for the quantities given are quite complicated. Closed form expressions for higher-order moments were recently derived in Nemirovsky [2013].

Many other quantities can be calculated using similar approaches. For example, we recently used this approach to develop an exact method for computing the expected age of an allele and its variance, which is implemented in `WFES` and is fully described in de Sanctis and de Koning [2016]. As other quantities are added in future versions of `WFES` they will be fully documented in the online documentation on Github.

1.2 Rapid solution of restricted linear systems

Significant computational savings occur when assuming that we know i. For example, it is commonly assumed that $i = 1$ or, equivalently, that the mutant enters the population as a single copy. In this case, the above calculations require only the first row of $\mathbf{N}$. This simplifies our computation considerably, because instead of computing an entire matrix inverse, we can instead just solve the linear system

$$(I - Q)^T\mathbf{N_1} = I_1 \tag{11}$$

for $\mathbf{N_1}$, where I_1 is the first column of the identity matrix. We solve this system by first obtaining a decomposition of $(I - Q)^T$.

We do have to calculate the entire B matrix in order to obtain conditional times to absorption. Fortunately, B only has two columns. We solve for the first by considering another system of linear equations

$$(I - Q)B_1 = R_1 \tag{12}$$

where B_1 is the first column of B and R_1 is the first column of R. Given that the decomposition of $(I - Q)^T$ was obtained in solving for $\mathbf{N_1}$, the solution to this system is trivial.

Since the Wright-Fisher model only contains two absorbing states, we can now compute

$$B_2 = \mathbf{1} - B_1 \tag{13}$$

where $\mathbf{1}$ is a vector of 1s and B_2 is the second column of B. Then $B_{1,1}$ is the probability of extinction and $B_{1,2}$ is the probability of fixation given that we started with a single copy. We can now use equation 7 to compute the expected time to fixation, given that fixation occurs, and likewise for extinction.

1.3 Parameterization of the Wright-Fisher model

The Wright-Fisher model describes the time-evolution of a bi-allelic locus in a population of fixed size with N_e individuals and non-overlapping generations. In its standard form, the model describes the number of mutant alleles in the next generation as a binomial draw from the number in the current generation. Assuming a diploid population with $2N_e$ chromosomes,

$$P_{i,j} = \binom{2N_e}{j}(\psi_i)^j(1 - \psi_i)^{2N_e - j}, \tag{14}$$

where ψ_i is the probability of being a mutant allele in the next generation given i alleles in the current generation, and $P_{i,j}$ expresses the chance of the population moving from i to j copies of a mutant allele within one generation.

Following Ewens [2004], we parameterize the fitness of diploid individuals as follows, where a is the wild-type state and A the mutant:

Genotype	Fitness
AA	$1+s$
Aa	$1+sh$
aa	1

Given this fitness model and bi-directional mutation, the corresponding transition probability matrix can be expressed using the above formula for $P_{i,j}$ and

$$\psi_i = \frac{\big[(1+s)i^2 + (1+sh)i(1-i)\big](1-u) + \big[(1+sh)i(1-i) + (1-i)^2\big]v}{(1+s)i^2 + 2(1+sh)i(1-i) + (1-i)^2} \tag{15}$$

where v is the forward mutation rate, u the backward mutation rate, s the selection coefficient, and h the dominance coefficient.

1.3.1 Haploid model

For reference, we also implemented an analogous haploid model (with N_e chromosomes), including selection and bi-directional mutation:

$$P_{i,j} = \binom{N_e}{j}(\psi_i')^j(1-\psi_i')^{N_e-j}, \tag{16}$$

$$\psi_i' = \frac{i(s+1)(1-u) + (N_e-i)v}{i(s+1) + N_e - i} \tag{17}$$

1.4 Rapid calculation of the transition matrix

To efficiently calculate the Wright-Fisher transition probability matrix, we note the following recurrence relation for each row based on equation 14

$$p_{i,0} = (1-\psi_i)^{2N_e} \tag{18}$$

$$p_{i,j} = p_{i,j-1} \cdot \frac{2N_e - j + 1}{j} \cdot c_i \tag{19}$$

where ψ_i and $c_i = \psi_i/(1-\psi_i)$ must only be computed once for each row. Since each row of the matrix is independent of the others, rows can be calculated in parallel. To do this efficiently, we assign blocks of rows to different threads and store the collected results in a sparse matrix data structure.

2 Supplementary Results

2.1 Comparison of exact and approximate results

Supplementary Figure 1 shows how the probability of fixation changes under strong selection for the exact method (WFES), forward simulation (WF Monte Carlo), and under several diffusion approximations owing to Kimura. In this figure, as in the main text,"Kimura diffusion" refers to Kimura's analytic probability of fixation [Kimura, 1962] ignoring mutation and dominance in a diploid population:

$$P_{\text{Fix}}^{(1)} = \frac{1-e^{-s}}{1-e^{-2N_e s}} \tag{20}$$

assuming that the initial number of copies of the mutant allele is $p = 1/(2N_e)$. Similarly, "Kimura diffusion (weak selection)" refers to the usual further approximation of this result assuming small s:

$$P_{\text{Fix}}^{(2)} \approx \frac{s}{1-e^{-2N_e s}} \tag{21}$$

Note that the form of both of these equations is based upon the parameterization of the Wright-Fisher model described above (Supplementary Methods section 1.3), which differs slightly from the form used by Kimura.

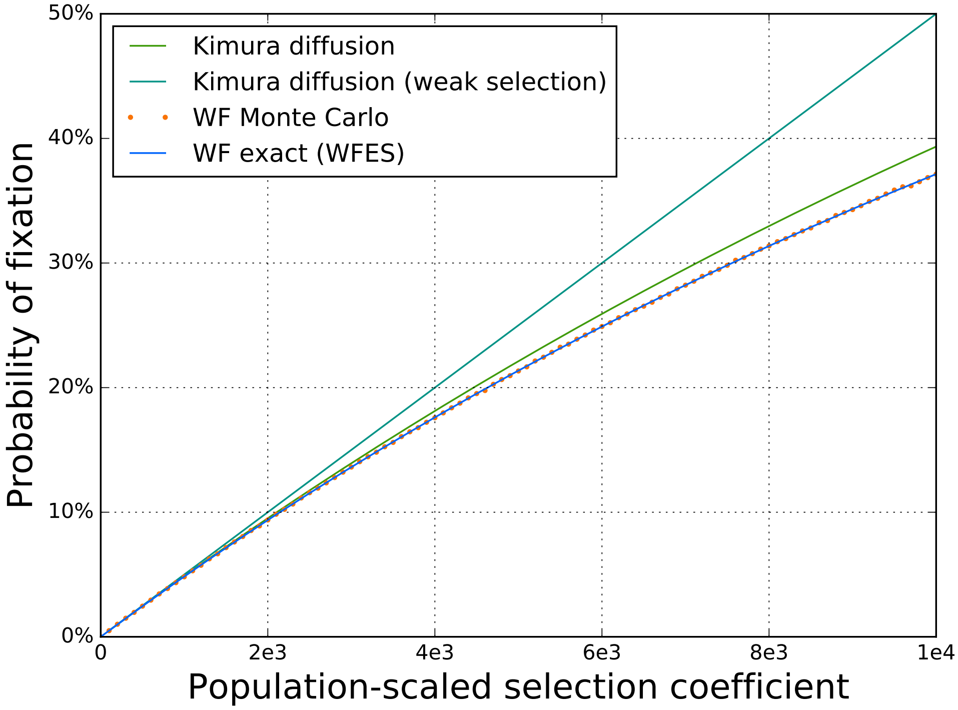

Figure 1: Probability of fixation for strongly selected alleles in a population of $N_e = 10,000$.

2.2 Effect of truncation

Each row of the Wright-Fisher matrix is essentially a binomial distribution. By cutting off the long near-zero tails of the distribution, it is possible to increase the sparsity of the system 2 (Supplementary Figure 2), thus lowering both the CPU and RAM requirements.

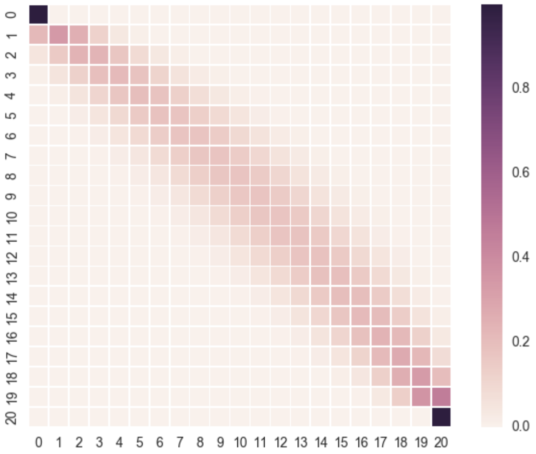

Figure 2: A small $N = 10$ WF transition matrix. The color of each cell represents the probability of transition for i to j copies within a single generation. Note the sparsity of the matrix.

We investigated the effects of different threshold values (ϵ) on the accuracy of the solver in Supplementary Table 1.

A threshold of $1e-20$ appears to provide good accuracy and performance for reasonably large population sizes. We note that for larger population sizes (i.e. above $N = 100,000$), the threshold value will have to be lowered to obtain exact solutions (within machine precision) on most current workstation computers.

Table 1: `WFES` performance with truncation, N_e=50,000. Benchmarked on a 16-core Intel Xeon CPU.

Epsilon	Relative error%	Memory usage, GB	Runtime, s
$1e-25$	0*	27.8	122.268
$1e-20$	0*	24.5	110.040
$1e-15$	0*	20.7	102.372
$1e-10$	0.02	16.1	80.684
$1e-09$	69.90	12.5	75.644

*Zero at machine precision.

2.3 Understanding the performance of WFES

Here we include additional results and details, which help to explain the computational advantages of `WFES`.

The intent of our approach is to make computations feasible by only calculating what we need, and by avoiding computing full, dense matrices. This is to reduce both time and space complexity, which we more fully explain below. One of the main problems with calculating the full matrix inverse (equation 2) is that it will be a dense matrix regardless of the sparsity of the underlying transition matrix. However, this full matrix is almost never required (see Section 1.2).

2.3.1 Definition of sparsity

When matrices are sparse, both time and space complexity can be reduced by only storing and working with the non-zero entries. Sparse linear algebra routines can then be used [Schenk et al., 2001], which take advantage of sparsity to avoid unnecessary computations. This can also have a very favourable impact on memory usage, as we explain in the next section.

Unless a non-zero value of ϵ is used in WFES, sparse elements of the transition matrix are defined as those that are evaluated to be zero *at machine precision*. Assuming 64-bit double precision floating point numbers, this means that with $\epsilon = 0$ an effective value of $\epsilon = 2.22 \times 10^{-308}$ is implied. For this reason, we emphasize that our meaning of "exact" in WFES applies without caveats to the symbolic linear algebra solutions presented in this supplement, which require no assumptions other than those made by the model itself. In our implementation, however, the solutions are *subject to machine precision*.

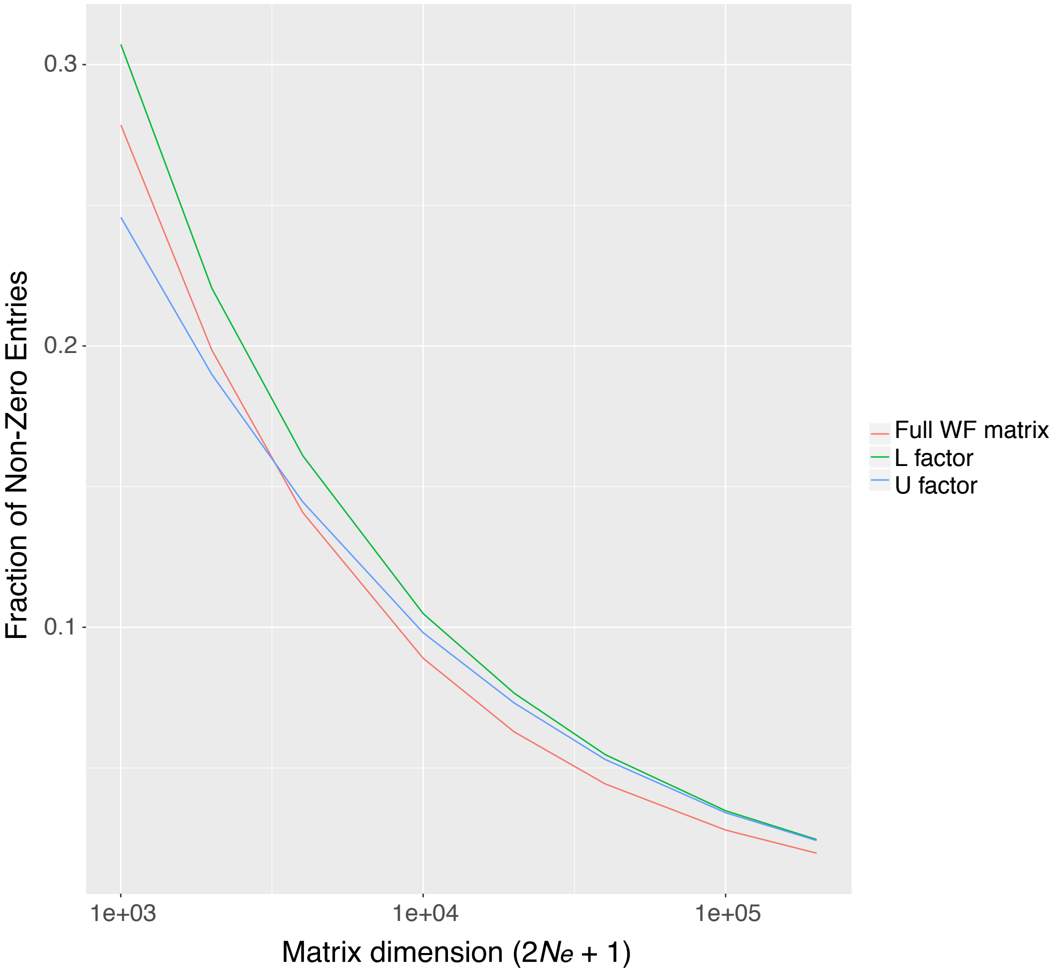

Figure 3: Percent non-zero entries for neutral Wright-Fisher transition matrices and their LU factors. Sparsity is defined as the fraction of zero entries (i.e., 100% $- fnz$, given the fraction of non-zero entries, fnz).

2.3.2 Space complexity reductions from sparsity

As can be seen in Supplemental Figure 3 (red), as Wright-Fisher transition matrices grow in size (for increasing effective population sizes), their sparsity generally increases. For a population size of $N_e = 100,000$, the fraction of non-zero entries in the transition matrix is only about 2% of the total number of entries. For a matrix of this size, storing the entire transition matrix at double precision would require 298GB of RAM ($200,001 \times 64/8$ bytes). Contrariwise, to store only the non-zero entries, we need $<$ 6 GB (plus some additional memory for indexing each non-zero entry).

Importantly, not only is the transition matrix of Wright-Fisher models sparse, but so too are the LU factors of the transition matrix (Supplemental Figure 3 green, blue), which are used to make all of the fast computations in WFES. As shown in Supplemental Figure 3, the LU factors of the transition matrix are about as sparse as the transition matrix itself. Thus, to get one row of the fundamental matrix, we only need to store the LU factors, which are each about 98% sparse (see Supplemental Figure 3; 5.96GB for the non-zero entries of each) together with a single row of $\mathbf{N}$ (1.5MB). As a result, to get a single row of the fundamental matrix, we need only about 11.9GB (75X less memory compared to storing the full transition matrix together with the full L and U matrices). Further, additional rows come at a cost of a mere 1.5MB each. This reduction is very favourable for scalability with large effective population sizes.

2.3.3 Peak RAM usage

WFES also makes use of some intermediate data structures and thus its peak memory usage can be greater than is implied above. Supplemental Figure 4 shows peak memory usage for increasing effective population sizes. For effective population sizes around $N_e = 100,000$, WFES uses a maximum of 74.1GB of RAM. As a general rule of thumb, workstations or servers with 100GB of RAM should be able to comfortable perform any WFES calculations for $N_e \leq 100,000$.

2.3.4 Runtime of component calculations

To show how the compute times of different phases in the WFES calculations scale with increasing effective population size, Supplemental Figure 5 shows the wall-clock time required for each phase of the calculations run on a 16-core Intel workstation. As can be seen in the figure, the symbolic factorization step (to reduce fill-in for the LU factors) takes the most time, with the calculation of the transition matrix itself requiring slightly less. The final construction of the LU factors (the 'numeric factorization' step) takes the next greatest amount of time, while the solution of the linear systems presented in Section 1 takes only a few seconds, even for large effective population sizes. In this example, the 'system solution' time is the time required to solve all linear systems needed to compute sojourn times, and fixation and extinction probabilities and times. Calculating allele age and other values can increase this time by a second or two for very large population sizes.

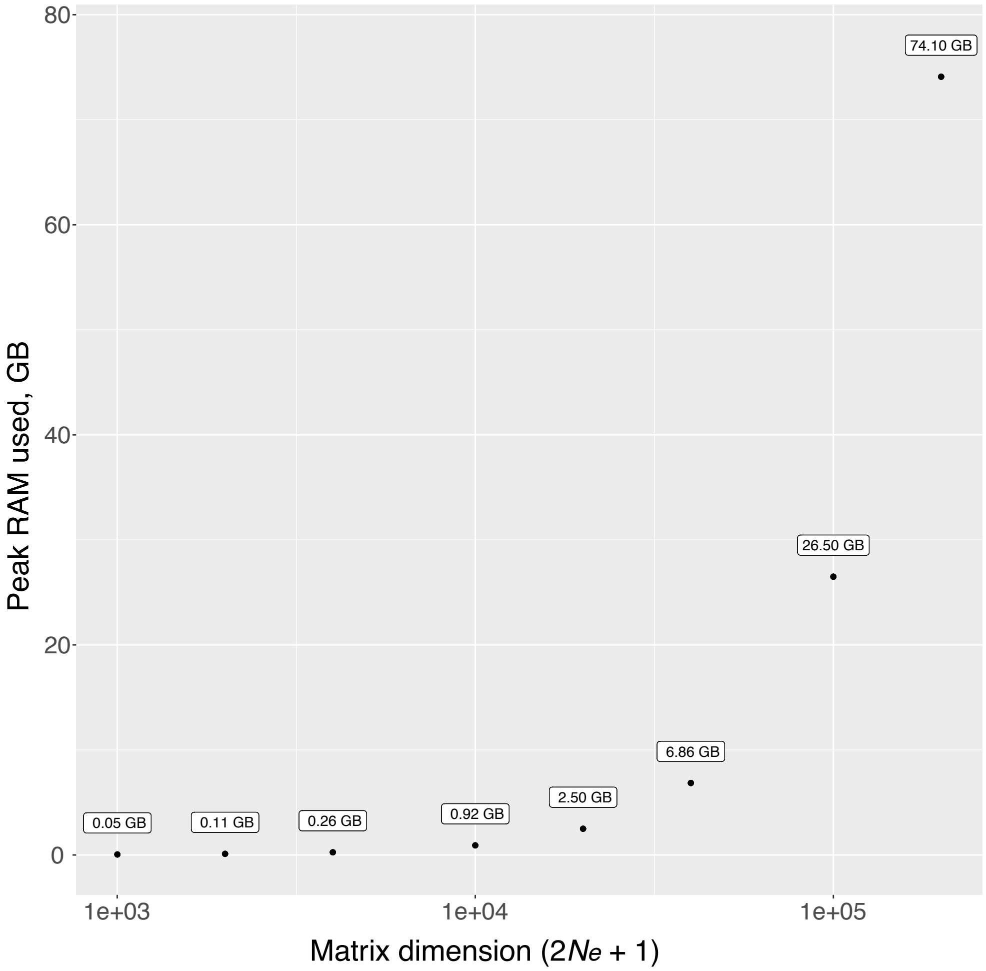

Figure 4: Peak total RAM used by WFES for runs with increasing effective population size.

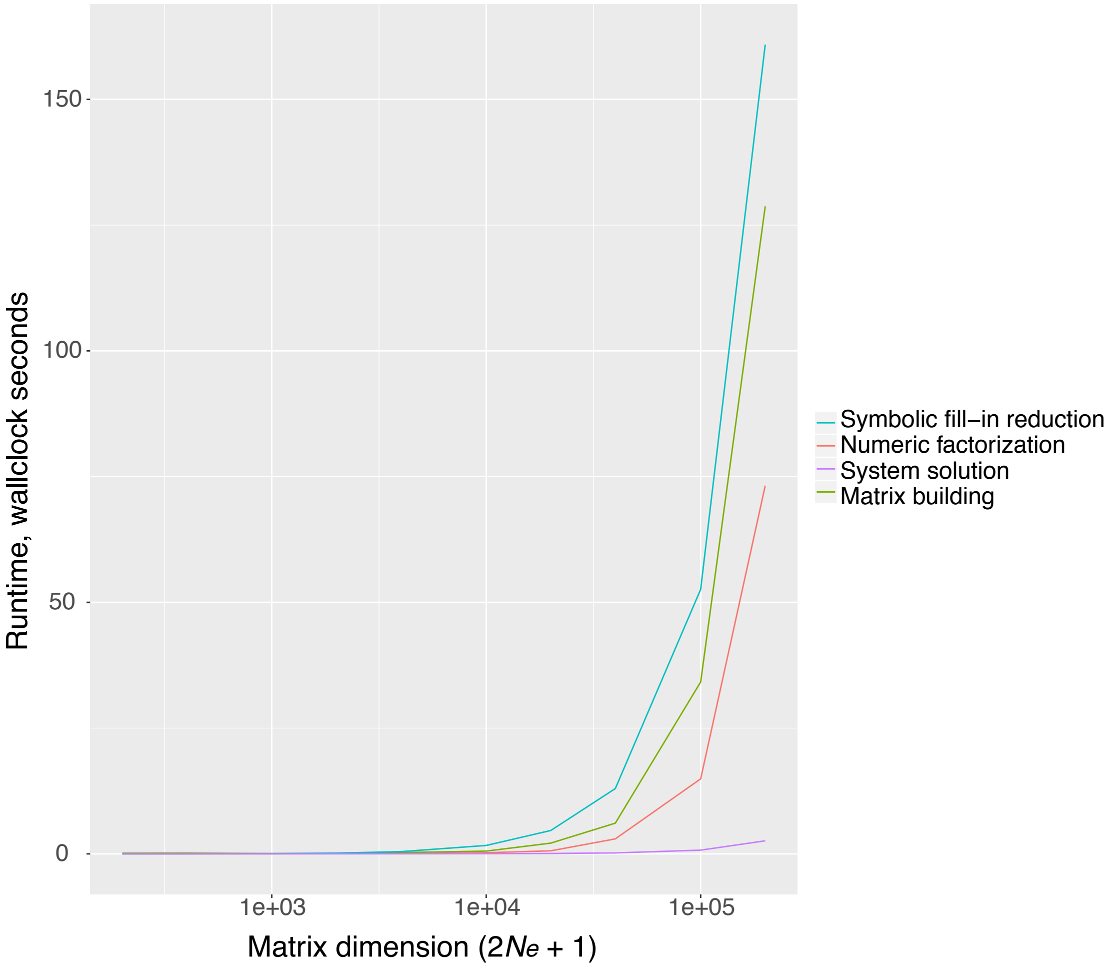

Figure 5: Total runtime (wallclock) required for different phases of the `WFES` computations. Note that the solution of linear systems takes merely seconds, even for large effective population sizes.

References

Intel, Inc. *Intel Math Kernel Library*, 2016.

O. Schenk, K. Gartner, and W. Fichtner. Efficient Sparse LU Factorization with Left-Right Looking Strategy on Shared Memory Multiprocessors. *BIT Numerical Mathematics*, 40 (1):158–176, March 2000. ISSN 0006-3835, 1572-9125. doi: 10.1023/A:1022326604210.

Olaf Schenk, Klaus Gärtner, Wolfgang Fichtner, and Andreas Stricker. Pardiso: a high-performance serial and parallel sparse linear solver in semiconductor device simulation. *Future Generation Computer Systems*, 18(1):69–78, 2001.

John G. Kemeny and Laurie J. Snell. *Finite Markov Chains.* Undergraduate Texts in Mathematics. Springer, 1960. ISBN 0-387-90192-2.

Danil Nemirovsky. Tensor approach to mixed high-order moments of absorbing Markov chains. *Linear Algebra and its Applications*, 438(4):1900–1922, February 2013. ISSN 0024-3795. doi: 10.1016/j.laa.2011.08.027.

Bianca de Sanctis and A. P. Jason de Koning. An exact approach for rapid computation of the expected age of an allele and its variance. *Submitted*, 2016.

Warren Ewens. *Mathematical Population Genetics 1.* Springer International Publishing, 2 edition, 2004. ISBN ISBN 978-0-387-21822-9.

Motoo Kimura. On the probability of fixation of mutant genes in a population. *Genetics*, 47(6):713, 1962.