

## Links are disabled because this is a static copy of a profile report





total\_hamiltonian (Calls: 27, Time: 2.354 s)

Generated 15-Jul-2024 18:42:29 using performance time.

function in file D:\Aalto\2324\BScThesis\FullRepo\parallelsimulations\_finitebath\src\total\_hamiltonian.m

Copy to new window for comparing multiple runs

### Lines where the most time was spent

Line Number	Code	Calls	Total Time	% Time	Time Plot
42	<code>g = mutual*(triu(-(gamma/sqrt(...</code>	27	1.253 s	53.2%	
71	<code>H = [H1, lambda; lambda', w];</code>	27	0.601 s	25.5%	
48	<code>H1 = g+g';</code>	27	0.349 s	14.8%	
73	<code>end</code>	27	0.145 s	6.1%	
52	<code>H1(1:N+1:end) = omega_j;</code>	27	0.004 s	0.2%	
All other lines			0.003 s	0.1%	
Totals			2.354 s	100%	

### Function listing

time	Calls	line
		30 function H = total_hamiltonian (N, w, mutual, gamma, omega_j)
		31
		32 % BATH
		33
		34 % Constructs an upper triangular N-by-N matrix of uniformly distributed
		35 % random numbers between -(gamma/sqrt(N)) and (gamma/sqrt(N)).
		36 % It represents the normalized by 1/sqrt(N) coupling strength between the
		37 % spins on the bath, the off-diagonal elements of the Hamiltonian matrix.
		38 % Thanks to the symmetry of the coupling between two spins and randomness,
		39 % it is enough to take only the upper triangular matrix.
		40 % g is relevant only if internal coupling is to be considered for the bath
		41 % model (mutual=1), otherwise (mutual=0) it becomes zero.
1.253	27	42 g = mutual*(triu(-(gamma/sqrt(N)) + 2*(gamma/sqrt(N))*rand(N),1));
		43
		44 % Constructs a symmetric matrix of coupling strengths, by taking the sum of
		45 % the upper triangular coupling strength matrix and its transpose. The
		46 % (Hermitian this way) Hamiltonian is constructed with diagonal elements
		47 % being zero.
0.349	27	48 H1 = g+g';
		49
		50 % Correct the Hamiltonian by replacing its diagonal elements with the
		51 % energy gaps of the spins of the bath.
0.004	27	52 H1(1:N+1:end) = omega_j;
		53
		54 % Diagonalize the bath Hamiltonian. vek1 is a matrix with column
		55 % eigenvectors and ek1 is a diagonal matrix of eigenvalues.

```

56 % [vek1, ek1] = eig(H1);
57
58 % BATH AND QUBIT
59
60 % Generates a column vector of uniformly distributed random numbers
61 % between -(gamma/sqrt(N)) and (gamma/sqrt(N)). It represents the
62 % normalized by 1/sqrt(N) coupling strength between the qubit and the
63 % spins on the bath.
64 % That sets the coupling between the spins and the coupling of the qubit
65 % with the spins at the same level.
0.003 27 66 lambda = -(gamma/sqrt(N)) + 2*(gamma/sqrt(N))*rand(N,1);
67
68 % Build the total Hamiltonian by concatenating lambda and its transpose
69 % (due to symmetry) as the last (N+1) column and last (N+1) row, while the
70 % last diagonal element is the frequency of the qubit.
0.601 27 71 H = [H1, lambda; lambda', w];
72
0.145 27 73 end

```