http://www.fftw.org/

FFTW

Download GitHub Mailing List Benchmark Features Documentation FAQ Links Feedback

Introduction

FFTW is a C subroutine library for computing the discrete Fourier transform (DFT) in one or more dimensions, of arbitrary input size, and of both real and complex data (as well as of even/odd data, i.e. the discrete cosine/sine transforms or DC applications.

The latest official release of FFTW is version **3.3.8**, available from our download page. Version 3.3 introduced support for the AVX x86 extensions, a distributed-memory implementation on top of MPI, and a Fortran 2003 API. Version 3.3.1 int

The FFTW package was developed at MIT by Matteo Frigo and Steven G. Johnson.

Our benchmarks, performed on on a variety of platforms, show that FFTW's performance is typically superior to that of other publicly available FFT software, and is even competitive with vendor-tuned codes. In contrast to vendor-tuned codes modification. Hence the name, "FFTW," which stands for the somewhat whimsical title of "**Fastest Fourier Transform in the West**."

Subscribe to the fftw-announce mailing list to receive release announcements (or use the web feed).

The forward (`FFTW_FORWARD`) discrete Fourier transform (DFT) of a 1d complex array X of size n computes an array Y, where:

$$Y_k = \sum_{j=0}^{n-1} X_j e^{-2\pi jk\sqrt{-1}/n} .$$

The backward (`FFTW_BACKWARD`) DFT computes:

$$Y_k = \sum_{j=0}^{n-1} X_j e^{2\pi jk\sqrt{-1}/n} .$$

http://www.fftw.org/

FFTW

Download GitHub Mailing List Benchmark Features Documentation FAQ Links Feedback

Introduction

FFTW is a C subroutine library for computing the discrete Fourier transform (DFT) in one or more dimensions, of arbitrary input size, and of both real and complex data (as well as of even/odd data, i.e. the discrete cosine/sine transforms or DC applications.

The latest official release of FFTW is version **3.3.8**, available from our download page. Version 3.3 introduced support for the AVX x86 extensions, a distributed-memory implementation on top of MPI, and a Fortran 2003 API. Version 3.3.1 int

The FFTW package was developed at MIT by Matteo Frigo and Steven G. Johnson.

Our benchmarks, performed on on a variety of platforms, show that FFTW's performance is typically superior to that of other publicly available FFT software, and is even competitive with vendor-tuned codes. In contrast to vendor-tuned codes modification. Hence the name, "FFTW," which stands for the somewhat whimsical title of "**Fastest Fourier Transform in the West**."

Subscribe to the fftw-announce mailing list to receive release announcements (or use the web feed).

The forward (`FFTW_FORWARD`) discrete Fourier transform (DFT) of a 1d complex array X of size n computes an array Y, where:

$$Y_k = \sum_{j=0}^{n-1} X_j e^{-2\pi jk\sqrt{-1}/n} .$$

The backward (`FFTW_BACKWARD`) DFT computes:

$$Y_k = \sum_{j=0}^{n-1} X_j e^{2\pi jk\sqrt{-1}/n} .$$

Downloading FFTW

Mailing list / Announcements

Subscribe to the fftw-announce mailing list on Google Groups to receive an email when FF

You can contact the FFTW authors at fftw@fftw.org.

FFTW 3.3.8

Version 3.3.8 is the latest stable release of FFTW, and full source code is found here:

- **http:** fftw-3.3.8.tar.gz (**ftp:** fftw-3.3.8.tar.gz) (md5sum) (4.1MB)
- you can also browse the ftp site

http://www.fftw.org/

FFTW

Download GitHub Mailing List Benchmark Features Documentation FAQ Links Feedback

Introduction

FFTW is a C subroutine library for computing the discrete Fourier transform (DFT) in one or more dimensions, of arbitrary input size, and of both real and complex data (as well as of even/odd data, i.e. the discrete cosine/sine transforms or DC applications.

The latest official release of FFTW is version **3.3.8**, available from our download page. Version 3.3 introduced support for the AVX x86 extensions, a distributed-memory implementation on top of MPI, and a Fortran 2003 API. Version 3.3.1 int

The FFTW package was developed at MIT by Matteo Frigo and Steven G. Johnson.

Our benchmarks, performed on on a variety of platforms, show that FFTW's performance is typically superior to that of other publicly available FFT software, and is even competitive with vendor-tuned codes. In contrast to vendor-tuned codes modification. Hence the name, "FFTW," which stands for the somewhat whimsical title of "**Fastest Fourier Transform in the West**."

Subscribe to the fftw-announce mailing list to receive release announcements (or use the web feed).

The forward (`FFTW_FORWARD`) discrete Fourier transform (DFT) of a 1d complex array X of size n computes an array Y, where:

$$Y_k = \sum_{j=0}^{n-1} X_j e^{-2\pi jk\sqrt{-1}/n} .$$

The backward (`FFTW_BACKWARD`) DFT computes:

$$Y_k = \sum_{j=0}^{n-1} X_j e^{2\pi jk\sqrt{-1}/n} .$$

Downloading FFTW

Mailing list / Announcements

Subscribe to the fftw-announce mailing list on Google Groups to receive an email when FF

You can contact the FFTW authors at fftw@fftw.org.

FFTW 3.3.8

Version 3.3.8 is the latest stable release of FFTW, and full source code is found here:

- **http:** fftw-3.3.8.tar.gz (**ftp:** fftw-3.3.8.tar.gz) (md5sum) (4.1MB)
- you can also browse the ftp site

```
./configure
make
make install
```

http://www.fftw.org/fftw3_doc/Fortran-Examples.html

```
double complex in, out
dimension in(N), out(N)
integer*8 plan

call dfftw_plan_dft_1d(plan,N,in,out,FFTW_FORWARD,FFTW_ESTIMATE)
call dfftw_execute_dft(plan, in, out)
call dfftw_destroy_plan(plan)
```

http://www.fftw.org/fftw3_doc/Fortran-Examples.html

```
double complex in, out
dimension in(N), out(N)
integer*8 plan

call dfftw_plan_dft_1d(plan,N,in,out,FFTW_FORWARD,FFTW_ESTIMATE)
call dfftw_execute_dft(plan, in, out)
call dfftw_destroy_plan(plan)
```

FFTW_MEASURE

http://www.fftw.org/fftw3_doc/Fortran-Examples.html

```
double complex in, out
dimension in(N), out(N)
integer*8 plan

call dfftw_plan_dft_1d(plan,N,in,out,FFTW_FORWARD,FFTW_ESTIMATE)
call dfftw_execute_dft(plan, in, out)
call dfftw_destroy_plan(plan)
```

FFTW_MEASURE

Check normalization!

http://www.fftw.org/fftw3.pdf