Contribute to Math.NET Numerics

===============================

Math.NET Numerics is driven by the community and contributors like you. I'm excited that you're interested to help us move forward and improve Numerics. We usually accept contributions and try to attribute them properly, provided they keep the library consistent, focused and mathematically accurate. Have a look at the following tips to get started quickly. I'm looking forward to your pull requests! Thanks!

— \*Christoph Rüegg (@cdrnet)\*

## Getting Started

- Make sure you have a [GitHub account](https://github.com/signup/free), it's free.

- Please configure a proper name and email address in git ([how to](https://help.github.com/articles/set-up-git)). Real names are preferred, but it is acceptable to use an alias or even an obviously fake email address if you wish not to be contacted, as long as something is explicitly configured (not the default).

- Fork the [mainline repository](https://github.com/mathnet/mathnet-numerics) on GitHub ([how to](https://help.github.com/articles/fork-a-repo)).

We use the [Fork & Pull Model](https://help.github.com/articles/using-pull-requests/), as common for GitHub projects. If you've already contributed to another GitHub project then you're all set. If not, [here is another introduction](https://gun.io/blog/how-to-github-fork-branch-and-pull-request/).

\*\*C# Solutions, Projects and Files\*\*

We have two kind of C# projects: primary (\*Numerics.csproj, UnitTests.csproj\*) and secondary (\*Numerics-xy.csproj, UnitTests-xy.csproj\*). The primary ones are the common VisualStudio project files you usually work with. The secondary projects on the other hand are not intended to be modified and include all files automatically. Whenever you need to add, remove or move a file, please do so in the primary projects only. In most cases we recommend to work with the `MathNet.Numerics.sln` solution which only includes primary projects anyway - except when working on and testing portability/compatibility.

\*\*F# Projects\*\*

F# does not support the wildcard approach of the C# projects by design, so whenever you add, remove or move an F# file please manually update all F# projects accordingly, including the secondary platform specific ones in the `MathNet.Numerics.All.sln` solution. This is a bit tedious but we have not found a better solution yet.

\*\*Separate Branch per Pull Request\*\*

We recommend that you create a separate branch for each pull request, as opposed to using master. This makes it much easier to continue working on a pull request even after it has been opened on GitHub. Remember that GitHub automatically includes all future commits of the same branch to the pull request.

\*\*Focused\*\*

We prefer a couple small pull requests over a single large one that targets multiple things at once.

### When fixing a bug ...

If you have a good idea how to fix it, directly open a pull request. Otherwise you may want to open an issue first (at GitHub) and discuss it there. If you can reproduce the bug with simple enough code, please consider adding a Unit Test that fails to confirm the bug.

### When extending features ...

If you're extending some feature which is similar and close to existing code, for example adding a new probability distribution or a new Bessel-related special function, it's fine to directly open a pull request. We're likely to accept such pulls.

### When adding new features ...

If you intend to add completely new features, say some spatial routines for geometrical transformations, we recommend to [talk to us](https://discuss.mathdotnet.com/c/numerics) first. This is mostly to avoid wasting your time in case we decide not to accept it, or require major refactoring. If the features is quite small it is perfectly fine to just open a pull request though. Sometimes it's easier to just show code instead of lengthy explanations.

Note that your work does not need to be finished or complete at the time you open the pull request (but please do mention this in the message), as GitHub pull requests are an excellent tool to discuss an early prototype or skeleton before it is fully implemented.

### When you wish to contribute but do not know where to start ...

Issues marked with "up-for-grabs" should be good candidates for a first contribution, but you can start with whatever you wish. If you decide to work on an existing issue, consider to add a comment to mention you're working on it.

What works very well is to try to build something with real world data that uses Math.NET Numerics: you either end up with a nice example that we would love to include or refer to, or you run into things which are missing, unintuitive, broken or just a bit weird, which we'd love to hear about so we (or you?) can fix it.

Should you stumble on weird English grammar or wording please do fix it - most of the contributors are not native English speakers. That includes this document.

## What to Avoid

\*\*Code Reformatting and Refactoring:\*\*

Please avoid starting with a major refactoring or any code reformatting without talking to us first.

\*\*Breaking Compatibility:\*\*

We try to follow [semantic versioning](https://semver.org/), meaning that we cannot break compatibility until the next major version. Since Numerics intentionally permits straight access to raw algorithms, a lot of member declarations are public and thus cannot be modified. Instead of breaking compatibility, it is often possible to create a new better version side by side though and mark the original implementation as obsolete and scheduled for removal on the next major version.

\*\*Merges:\*\*

Please avoid merging mainline back into your pull request branch. If you need to leverage some changes recently added to mainline, consider to rebase instead. In other words, please make sure your commits sit directly on top of a recent mainline master.