# NDT Lab – Arrays Interface

## Introduction

The arrays interface is a suite of matlab functions that provide a general purpose and expandable GUI for obtaining,processing and visualising data from ultrasonic arrays. The purpose of the array interface suite of programmes and functions is twofold:

1. To provide a flexible GUI with sufficient flexibility that new processing functions and array controller drivers can be automatically incorporated without the authors of those functions having to alter or programme any part of the GUI.
2. To provide a standalone executable than can be used without Matlab by third parties.

## Philosophy

### Separation of GUI from processing code

The GUI can be updated without having to alter any processing code. For this to work, the interfaces between the GUI and other code have to be clearly (and flexibly) defined.

### GUI is composed of various standalone entities

This allows further independent improvements to be made. Most GUI components will be either panels (that can be slotted into any window) or figures. Each component is typically a single function file, which contains the layout, setup code as well as all the callback handling for that component.

### Expandability

The intention is that once written, the GUI does not need anything other than cosmetic updates. Instead, it will automatically expand and adapt to processing functions as they are added. The current favoured mechanism for this is to have one editable “list” function that provides details of each processing function to include in the GUI when it executes. The list function will contain details of (a) the processing function address and (b) the details of the input options from which a GUI control will automatically be generated. All processing functions must therefore have the same form:

result = fn\_processing\_function(exp\_data, options)

exp\_data will be the standard array data structure and options is a structured variable that contains (a) fields relating to process parameters (e.g. resolution, filtering), (b) the name of the function used to display the output (e.g. a 2D Cartesian display function) and any options associated with it, such as second level processing function handles (API, defect sizing, S-matrix extraction etc). \*maybe the input details should be somehow in the processing function and a standard option is sat option.return\_inputs that causes the function to just return the necessary data instead of the actual result?\*