# PACE

## *Project Management Board Meeting 11 Dec. 2020*

AGENDA

- Principal sub-projects – minimum viable product
- Technical update
- Project plan and progress against milestones
- Project finances

UKRI Science and Technology Facilities Council ISIS

# PACE Goals – three sub-projects, minimum viable product

- Interface to third party simulation codes
  - SpinW (now a separate project)
  - Phonon calculations (Euphonic)
  - API for generic user (Matlab, Python and C++), and 3rd party codes
- Optimisation and resolution convolution algorithms
  - Parallel Tobyfit
  - New approaches
- Performance and usability framework for
  - parallel and distributed computing architecture :
    - DAaaS, SCARF, but will also work on high end laptop/desktop
  - Matlab and Python user interface
  - Handling large datasets out of memory
  - Generic projections : user-defined

# ALC “SpinWCore / libMcPhase” Project

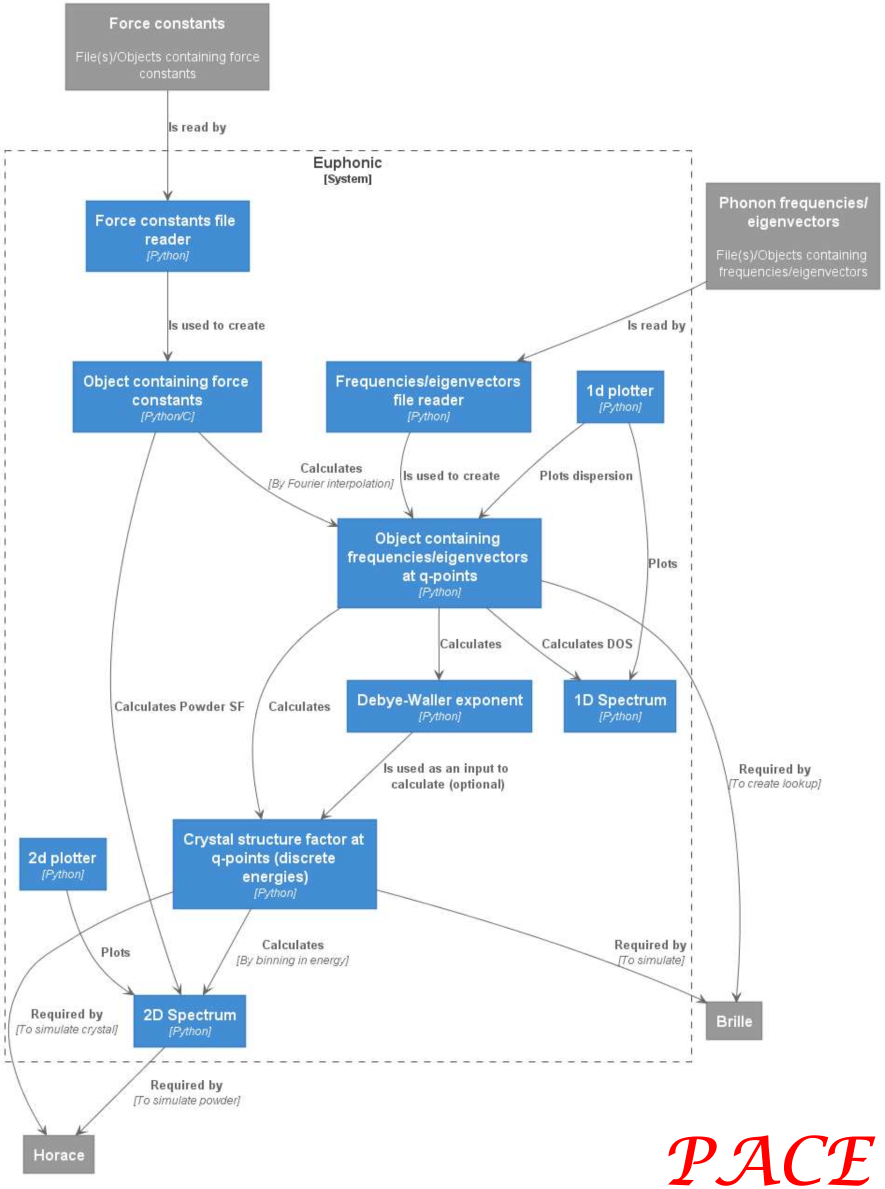
- Developer will start Dec 14. 2020
- Deliverables:
  - Parallelised compiled (C++) core computation routines for SpinW (rewrite of `spinwave.m`, 6 months)
  - Rewrite of core parts of McPhase as a C++ library with a Python interface (libmcphase, 18 months)
- Funding from Ada Lovelace Centre to Mar 2022 (15 months) then developer will be funded by ISIS.
- In scope:
  - Full CI for SpinW/SpinWCore and libMcPhase (both projects need new unit/system tests and servers setup)
- Not in scope: *[of the ALC project funding, not of PACE overall]*
  - Additional SpinW features (e.g. Python interface, user requests)
  - Integration with other PACE projects (Horace, Brille).

# SpinW specifics

- Currently unmaintained (lack of dedicated developer).
- Long term goal is to convert to Python.
  - Currently a functioning Python interface using compiled Matlab exists.
  - C++ SpinWCore might be basis for new Python version, but will need dedicated developer
- Urgently needs comprehensive set of unit / system tests and CI infrastructure (will be met by ALC project).
  - Could use cloud CI (e.g. Circle, Travis) but those only provide Matlab for Linux instances.
- Integration with PACE:
  - Existing Horace interface within SpinW codebase (will stay).
  - New Brille interface within SpinW codebase (now implemented by MDL).
- Many of new features asked for by users (multi-k structures, multi-magnon continuum, fitting powder data) but lacking developer effort.

# Euphonic: Since March 20

- Major refactor of Euphonic's API completed (more user friendly and maintainable, easy handling of units, can output any Euphonic object as .json file)
- Addition of spherical averaging tools for powder averaging (contributed by Abins developer Adam Jackson)
- More robust and user friendly command-line tools e.g. dispersion.py is now euphonic-dispersion
- Has been successfully validated against other computational codes
- Graduate effort (James King) – 3 month project improving testing
    - Migrating to Pytest
    - Adding Windows/Mac continuous integration nodes



Force constants

File(s)/Objects containing force constants

Is read by

Euphonic
[System]

Phonon frequencies/ eigenvectors

File(s)/Objects containing frequencies/eigenvectors

Force constants file reader
[Python]

Is used to create

Is read by

Object containing force constants
[Python/C]

Frequencies/eigenvectors file reader
[Python]

1d plotter
[Python]

Calculates
[By Fourier interpolation]

Is used to create

Plots dispersion

Object containing frequencies/eigenvectors at q-points
[Python]

Plots

Calculates

Calculates DOS

Calculates Powder SF

Calculates

Debye-Waller exponent
[Python]

1D Spectrum
[Python]

Is used as an input to calculate (optional)

Required by
[To create lookup]

2d plotter
[Python]

Crystal structure factor at q-points (discrete energies)
[Python]

Plots

Calculates
[By binning in energy]

Required by
[To simulate]

Required by
[To simulate crystal]

2D Spectrum
[Python]

Brille

Required by
[To simulate powder]

Horace

# Euphonic: Future Work

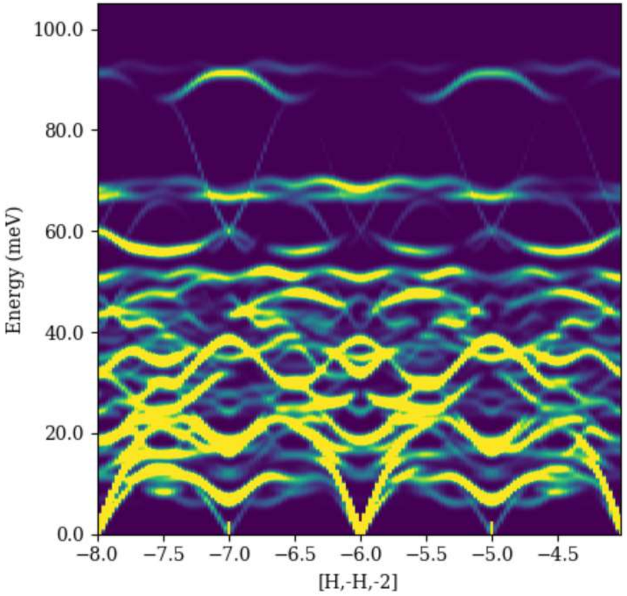
- Work towards publication on Euphonic

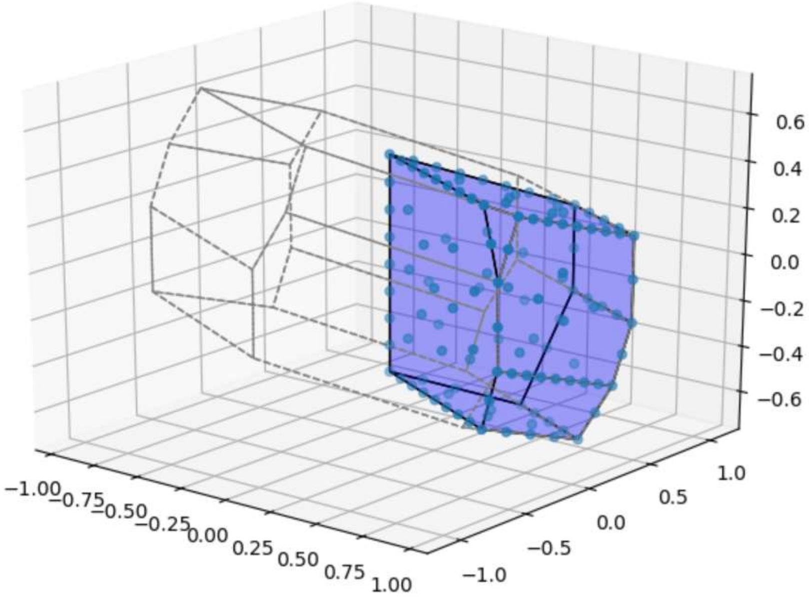
- Alongside publication aiming for:
  - 1.0.0 release of Euphonic
  - Release version of Euphonic-Horace-Interface

- Benchmark Euphonic performance against itself, CASTEP phonon tools, other codes etc: much has been done but it needs to be formalized

- Allow use of *brille* from Euphonic

- New integration repository for Horace-Euphonic interaction – Basis for testing, regressions





UKRI Science and Technology Facilities Council ISIS

PACE

# *brille* library and resolution

- Refactoring, handling improvements:
  - Proper handling of symmetry for interpolated vectors, matrices, and phonon eigenvectors
  - Equivalent mode caching
  - Some consequent performance impact, offset by….
  - Shared memory arrays

- Usability:
  - C++ library namespace encapsulation
  - Automated Python module build and publish pipeline
  - Automated hybrid Sphinx and Doxygen documentation

- AI with SpinW
  - Training time for neural nets reduced by order of magnitude to feasible size
  - Now 7000 cpu-hours

Keith T. Butler, Manh Duc Le, Jeyarajan Thiyagalingam, Toby G. Perring, "Interpretable, calibrated neural networks for analysis and understanding of inelastic neutron scattering data", https://arxiv.org/abs/2011.04584

- Staffing:
  - Greg Tucker (PDRA) leaving for ESS Lund, will continue to work on *brille* at a reduced level
  - Process of formalising this ongoing

# Tobyfit refactoring preparation

- Matlab instrument component and detector classes
    - rewritten and incorporated into a refactored Tobyfit.
    - Now part of new SQW object redesign (see framework, next slide)

- Parameter optimisation application
    - Design document finalised.
        - Generalises the fit functions, allows for different swappable core fitting engines.
        - Allows construction of more complex fit function
    - Can be implemented once the core SQW object rewrite is completed. The legacy Tobyfit version has been removed

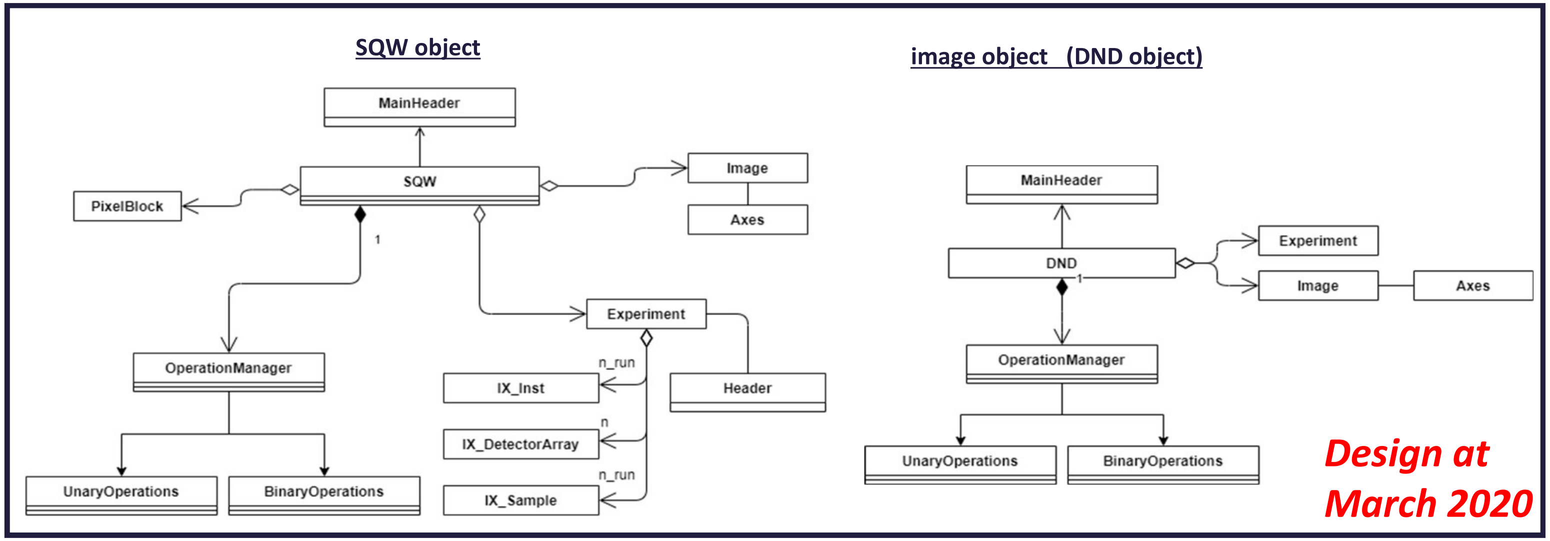
# Framework

- Support for large datasets (not fitting in memory) – implementation in progress
  - Dataset stored in temporary files, loaded into memory piece by piece for processing, with the results written back to file.
  - Requires conversion of large number of operations .
  - *Slows overall operation due to interaction with file system – price for working above memory limit. Algorithm-calling protocol ensures users know when this degradation is happening. 80% complete*
- MPI framework developed to support the parallelisation of a number of operations to significantly improve performance
  - Provides parallel inter-process communication through multiple frameworks, supporting file-based messages, the MATLAB Parallel toolbox and MPI framework
  - *Complete. Used to implement parallelization of SQW object generation (now in testing). Cut and symmetrize operations now planned*
- Compiled MATLAB *now available*
  - Compiled Matlab instances enable licence free operation e.g . from Python front-end
- Continuous Integration (CI) set up using ANVIL service - Captures cross-platform build and test functionality previously missing or executed by hand
  - Creates builds of Horace and Herbert for multiple MATLAB versions and multiple OS (initially Linux and Windows, extending to Mac, iDAaaS and SCARF). *Now allows coupling of non-master versions of Herbert and Horace (for new SQW, next slide).*
  - *Now also publishes documentation and release notes.*
  - *Version numbering more systematic, (=semantic versioning). Separately Euphonic and Brille are moving towards common semantic versioning, but Euphonic and Horace are not directly couped – coupling is between both of them and the Horace-Euphonic interface*

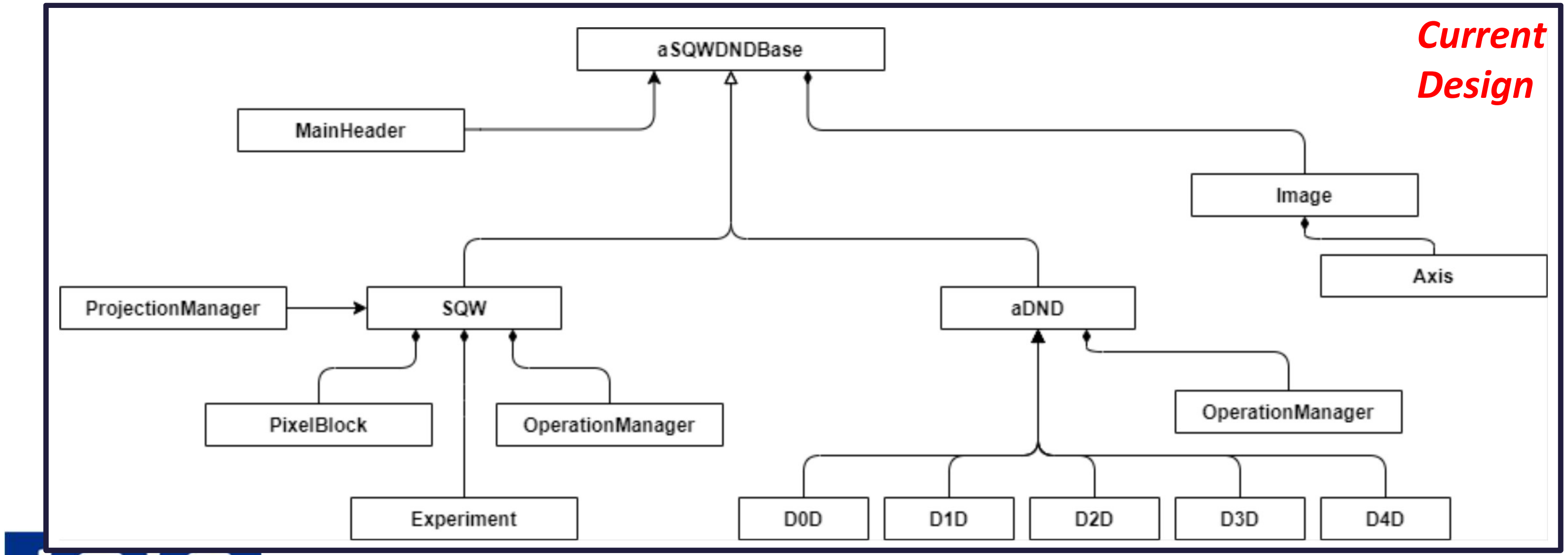
# Framework

## *SQW object redesign*

- Design for major update of the SQW object complete – implementation in progress
  - Supports interaction with the other PACE projects; eases future maintenance and development
  - Supports current functionality and known new requirements e.g. richer Instrument and Detector information required for resolution convolution.and compiled MATLAB
  - *Implementation found to be more complex due to interdependency between classes, and restarted as a complete rewrite to reduce inter-class coupling.*



*Design at March 2020*

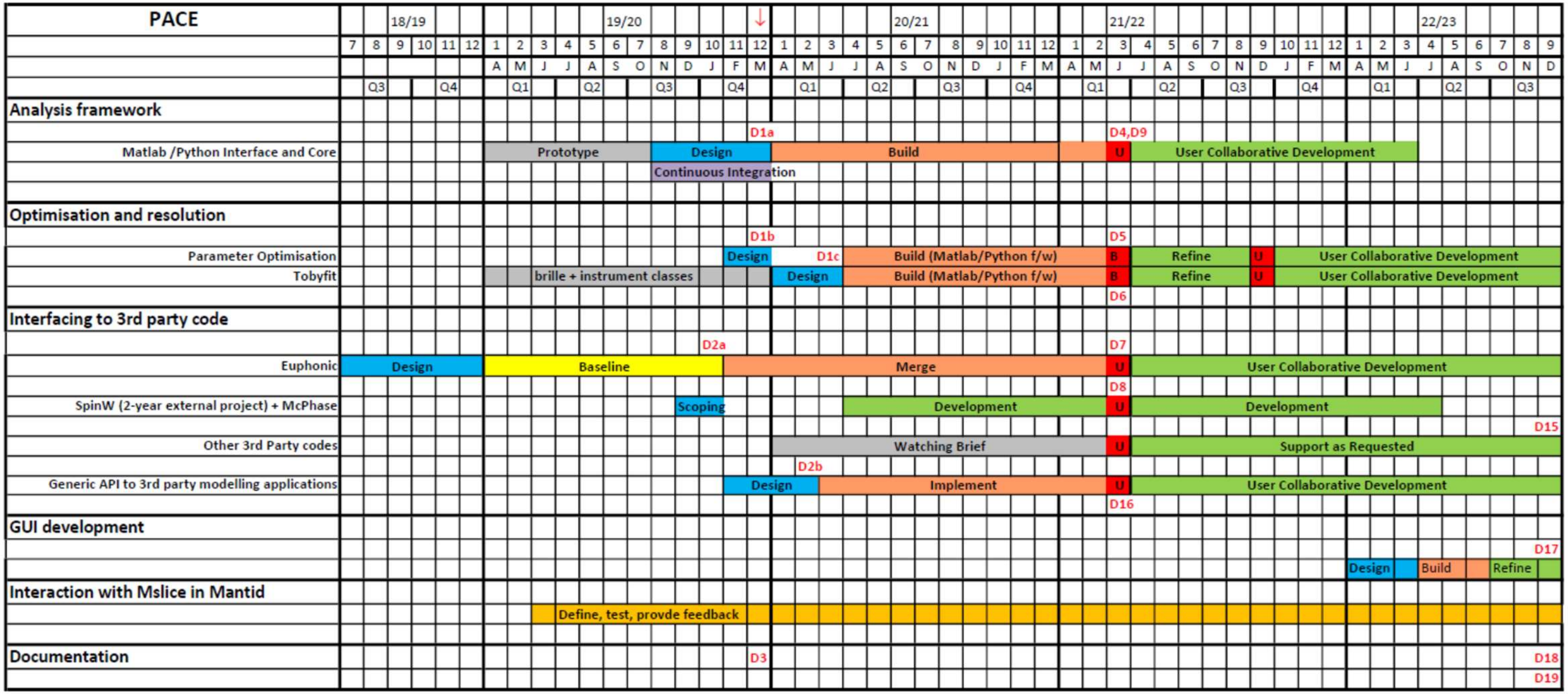


*Current Design*

Science and Technology Facilities Council

*PACE*

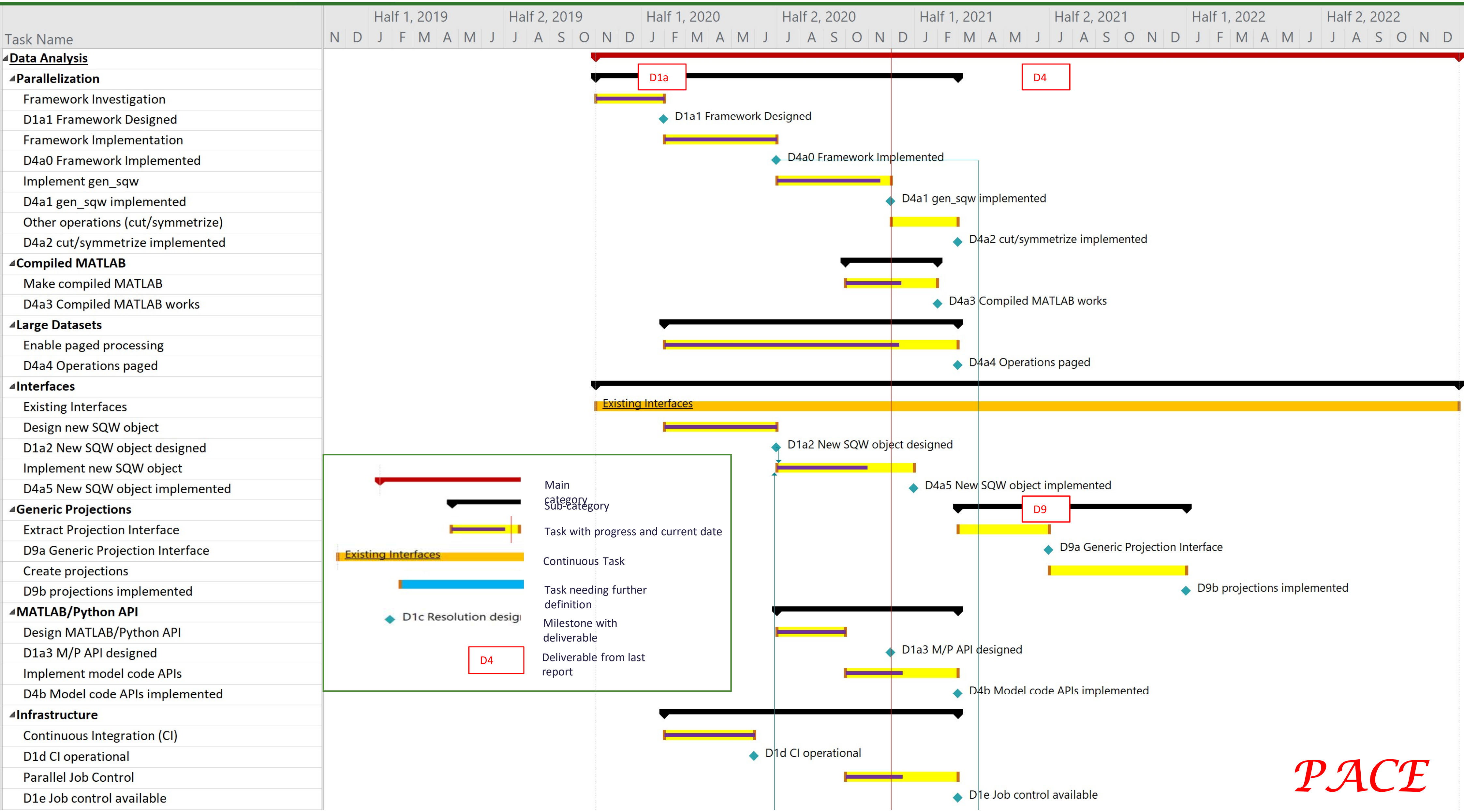
# Project plan



Key deliverbles are indicated by D1, D2 etc. Please refer to Project Management Plan

B = Beta release

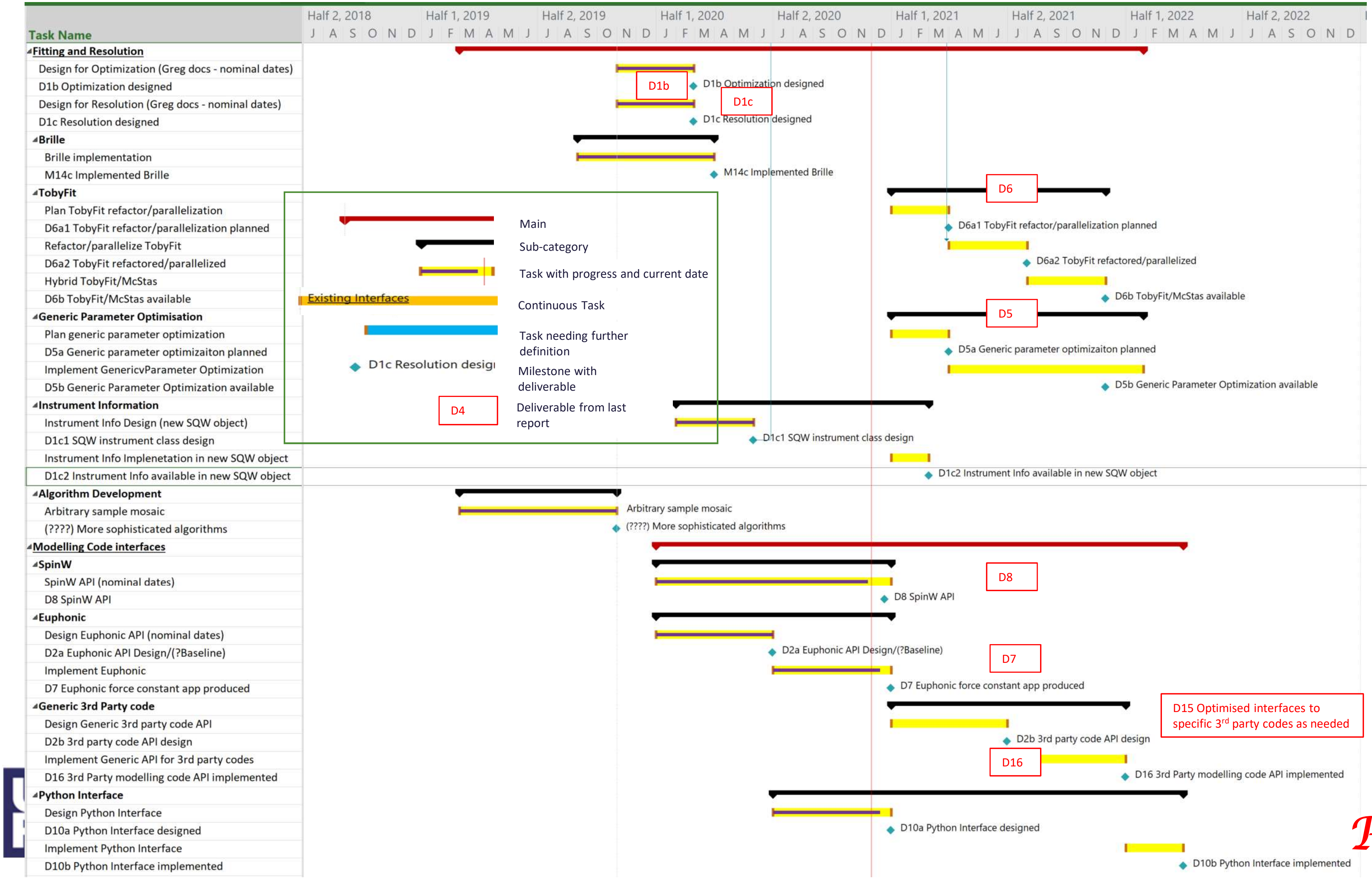
U = User release



Task Name

- **Data Analysis**
  - **Parallelization**
    - Framework Investigation
    - D1a1 Framework Designed
    - Framework Implementation
    - D4a0 Framework Implemented
    - Implement gen_sqw
    - D4a1 gen_sqw implemented
    - Other operations (cut/symmetrize)
    - D4a2 cut/symmetrize implemented
  - **Compiled MATLAB**
    - Make compiled MATLAB
    - D4a3 Compiled MATLAB works
  - **Large Datasets**
    - Enable paged processing
    - D4a4 Operations paged
  - **Interfaces**
    - Existing Interfaces
    - Design new SQW object
    - D1a2 New SQW object designed
    - Implement new SQW object
    - D4a5 New SQW object implemented
  - **Generic Projections**
    - Extract Projection Interface
    - D9a Generic Projection Interface
    - Create projections
    - D9b projections implemented
  - **MATLAB/Python API**
    - Design MATLAB/Python API
    - D1a3 M/P API designed
    - Implement model code APIs
    - D4b Model code APIs implemented
  - **Infrastructure**
    - Continuous Integration (CI)
    - D1d CI operational
    - Parallel Job Control
    - D1e Job control available

Half 1, 2019
Half 2, 2019
Half 1, 2020
Half 2, 2020
Half 1, 2021
Half 2, 2021
Half 1, 2022
Half 2, 2022
N D J F M A M J J A S O N D J F M A M J J A S O N D J F M A M J J A S O N D J F M A M J J A S O N D
D1a
D4
D1a1 Framework Designed
D4a0 Framework Implemented
D4a1 gen_sqw implemented
D4a2 cut/symmetrize implemented
D4a3 Compiled MATLAB works
D4a4 Operations paged
Existing Interfaces
D1a2 New SQW object designed
D4a5 New SQW object implemented
D9
D9a Generic Projection Interface
D9b projections implemented
D1a3 M/P API designed
D4b Model code APIs implemented
D1d CI operational
D1e Job control available
Main category
Sub-category
Task with progress and current date
Existing Interfaces
Continuous Task
Task needing further definition
D1c Resolution desig
Milestone with deliverable
D4
Deliverable from last report
PACE



**Task Name**

- **Fitting and Resolution**
  - Design for Optimization (Greg docs - nominal dates)
  - D1b Optimization designed
  - Design for Resolution (Greg docs - nominal dates)
  - D1c Resolution designed
  - **Brille**
    - Brille implementation
    - M14c Implemented Brille
  - **TobyFit**
    - Plan TobyFit refactor/parallelization
    - D6a1 TobyFit refactor/parallelization planned
    - Refactor/parallelize TobyFit
    - D6a2 TobyFit refactored/parallelized
    - Hybrid TobyFit/McStas
    - D6b TobyFit/McStas available
  - **Generic Parameter Optimisation**
    - Plan generic parameter optimization
    - D5a Generic parameter optimizaiton planned
    - Implement GenericvParameter Optimization
    - D5b Generic Parameter Optimization available
  - **Instrument Information**
    - Instrument Info Design (new SQW object)
    - D1c1 SQW instrument class design
    - Instrument Info Implenetation in new SQW object
    - D1c2 Instrument Info available in new SQW object
  - **Algorithm Development**
    - Arbitrary sample mosaic
    - (????) More sophisticated algorithms
- **Modelling Code interfaces**
  - **SpinW**
    - SpinW API (nominal dates)
    - D8 SpinW API
  - **Euphonic**
    - Design Euphonic API (nominal dates)
    - D2a Euphonic API Design/(?Baseline)
    - Implement Euphonic
    - D7 Euphonic force constant app produced
  - **Generic 3rd Party code**
    - Design Generic 3rd party code API
    - D2b 3rd party code API design
    - Implement Generic API for 3rd party codes
    - D16 3rd Party modelling code API implemented
  - **Python Interface**
    - Design Python Interface
    - D10a Python Interface designed
    - Implement Python Interface
    - D10b Python Interface implemented

Main

Sub-category

Task with progress and current date

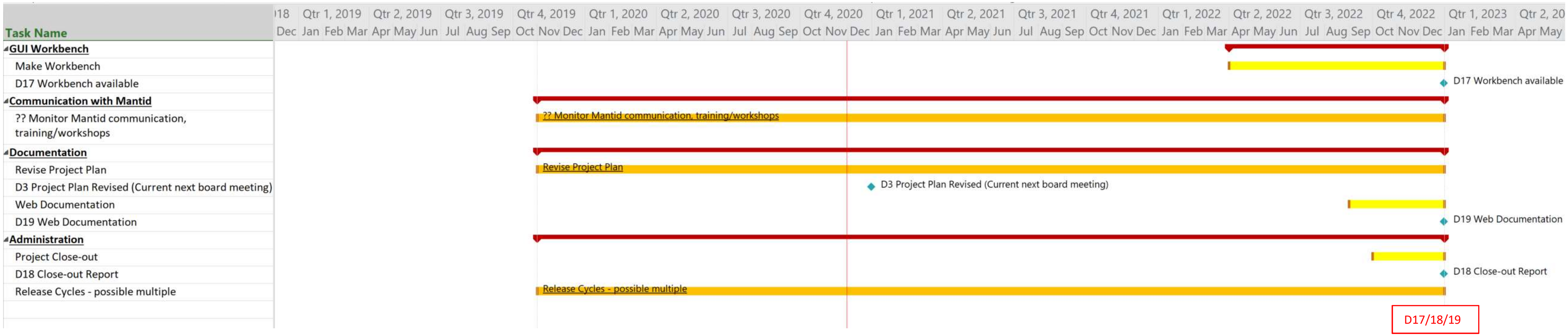
Continuous Task

Task needing further definition

Milestone with deliverable

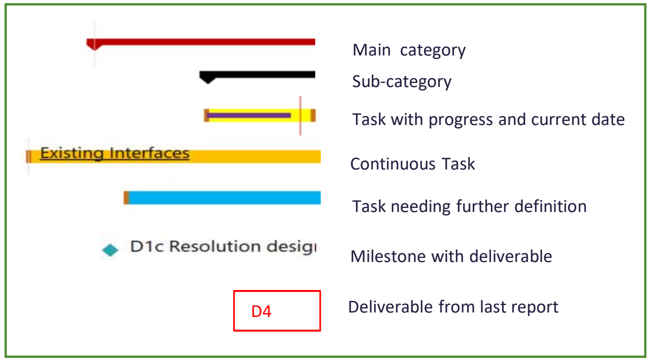
Deliverable from last report

*PACE*



| Task Name |
|---|
| **GUI Workbench** |
| Make Workbench |
| D17 Workbench available |
| **Communication with Mantid** |
| ?? Monitor Mantid communication, training/workshops |
| **Documentation** |
| Revise Project Plan |
| D3 Project Plan Revised (Current next board meeting) |
| Web Documentation |
| D19 Web Documentation |
| **Administration** |
| Project Close-out |
| D18 Close-out Report |
| Release Cycles - possible multiple |

)18 Qtr 1, 2019 Qtr 2, 2019 Qtr 3, 2019 Qtr 4, 2019 Qtr 1, 2020 Qtr 2, 2020 Qtr 3, 2020 Qtr 4, 2020 Qtr 1, 2021 Qtr 2, 2021 Qtr 3, 2021 Qtr 4, 2021 Qtr 1, 2022 Qtr 2, 2022 Qtr 3, 2022 Qtr 4, 2022 Qtr 1, 2023 Qtr 2, 20
Dec Jan Feb Mar Apr May Jun Jul Aug Sep Oct Nov Dec Jan Feb Mar Apr May Jun Jul Aug Sep Oct Nov Dec Jan Feb Mar Apr May Jun Jul Aug Sep Oct Nov Dec Jan Feb Mar Apr May Jun Jul Aug Sep Oct Nov Dec Jan Feb Mar Apr May
D17 Workbench available
?? Monitor Mantid communication, training/workshops
Revise Project Plan
D3 Project Plan Revised (Current next board meeting)
D19 Web Documentation
D18 Close-out Report
Release Cycles - possible multiple
D17/18/19

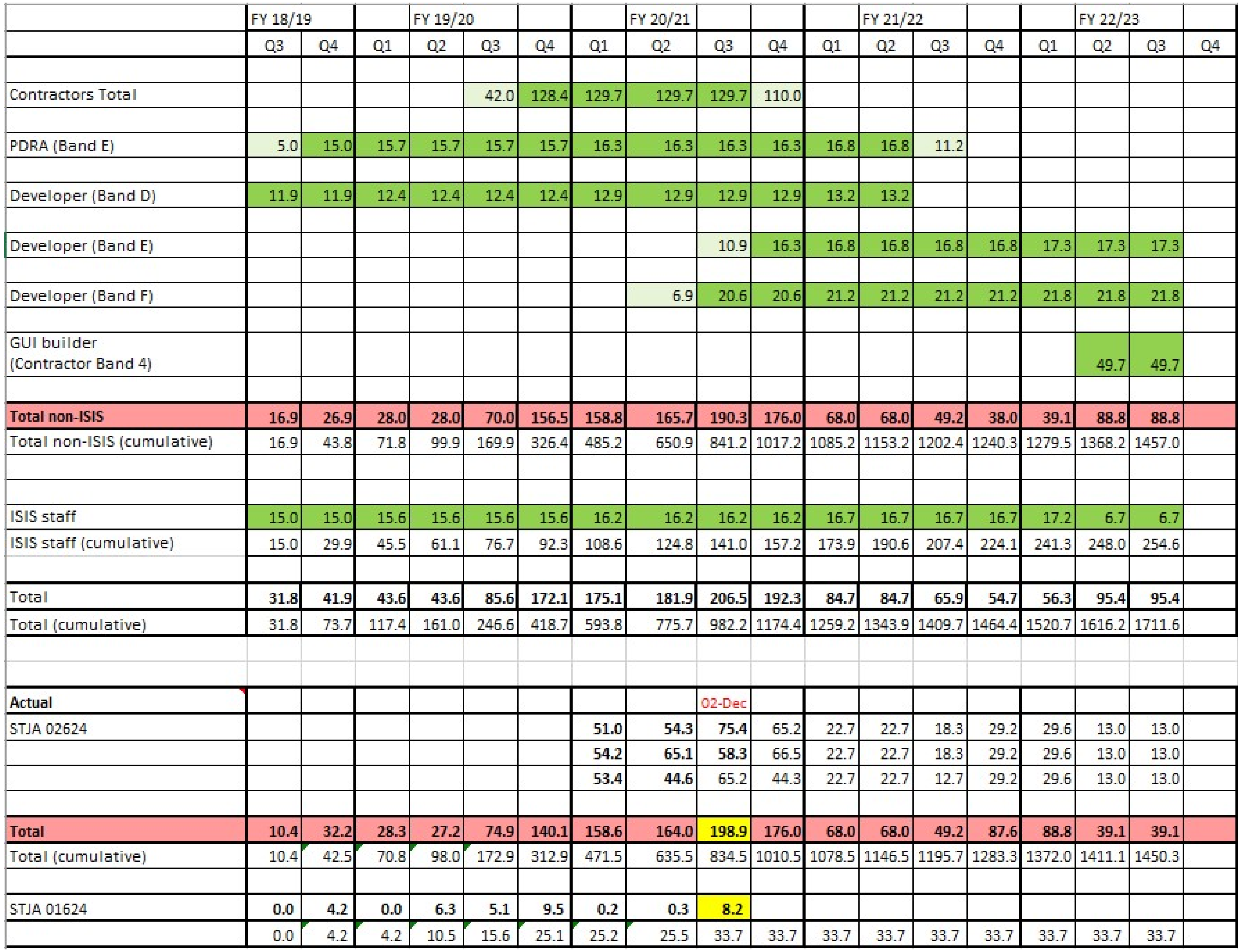

Main category
Sub-category
Task with progress and current date
Existing Interfaces
Continuous Task
Task needing further definition
D1c Resolution desig
Milestone with deliverable
D4
Deliverable from last report

Science and Technology Facilities Council

ISIS

*PACE*

# Project finances



| | FY 18/19 | | | FY 19/20 | | | | FY 20/21 | | | | FY 21/22 | | | | FY 22/23 | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | Q3 | Q4 | Q1 | Q2 | Q3 | Q4 | Q1 | Q2 | Q3 | Q4 | Q1 | Q2 | Q3 | Q4 | Q1 | Q2 | Q3 | Q4 |
| | | | | | | | | | | | | | | | | | | |
| Contractors Total | | | | | 42.0 | 128.4 | 129.7 | 129.7 | 129.7 | 110.0 | | | | | | | | |
| | | | | | | | | | | | | | | | | | | |
| PDRA (Band E) | 5.0 | 15.0 | 15.7 | 15.7 | 15.7 | 15.7 | 16.3 | 16.3 | 16.3 | 16.3 | 16.8 | 16.8 | 11.2 | | | | | |
| | | | | | | | | | | | | | | | | | | |
| Developer (Band D) | 11.9 | 11.9 | 12.4 | 12.4 | 12.4 | 12.4 | 12.9 | 12.9 | 12.9 | 12.9 | 13.2 | 13.2 | | | | | | |
| | | | | | | | | | | | | | | | | | | |
| Developer (Band E) | | | | | | | | | 10.9 | 16.3 | 16.8 | 16.8 | 16.8 | 16.8 | 17.3 | 17.3 | 17.3 | |
| | | | | | | | | | | | | | | | | | | |
| Developer (Band F) | | | | | | | | 6.9 | 20.6 | 20.6 | 21.2 | 21.2 | 21.2 | 21.2 | 21.8 | 21.8 | 21.8 | |
| | | | | | | | | | | | | | | | | | | |
| GUI builder (Contractor Band 4) | | | | | | | | | | | | | | | | 49.7 | 49.7 | |
| | | | | | | | | | | | | | | | | | | |
| **Total non-ISIS** | **16.9** | **26.9** | **28.0** | **28.0** | **70.0** | **156.5** | **158.8** | **165.7** | **190.3** | **176.0** | **68.0** | **68.0** | **49.2** | **38.0** | **39.1** | **88.8** | **88.8** | |
| Total non-ISIS (cumulative) | 16.9 | 43.8 | 71.8 | 99.9 | 169.9 | 326.4 | 485.2 | 650.9 | 841.2 | 1017.2 | 1085.2 | 1153.2 | 1202.4 | 1240.3 | 1279.5 | 1368.2 | 1457.0 | |
| | | | | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | | | | |
| ISIS staff | 15.0 | 15.0 | 15.6 | 15.6 | 15.6 | 15.6 | 16.2 | 16.2 | 16.2 | 16.2 | 16.7 | 16.7 | 16.7 | 16.7 | 17.2 | 6.7 | 6.7 | |
| ISIS staff (cumulative) | 15.0 | 29.9 | 45.5 | 61.1 | 76.7 | 92.3 | 108.6 | 124.8 | 141.0 | 157.2 | 173.9 | 190.6 | 207.4 | 224.1 | 241.3 | 248.0 | 254.6 | |
| | | | | | | | | | | | | | | | | | | |
| Total | **31.8** | **41.9** | **43.6** | **43.6** | **85.6** | **172.1** | **175.1** | **181.9** | **206.5** | **192.3** | **84.7** | **84.7** | **65.9** | **54.7** | **56.3** | **95.4** | **95.4** | |
| Total (cumulative) | 31.8 | 73.7 | 117.4 | 161.0 | 246.6 | 418.7 | 593.8 | 775.7 | 982.2 | 1174.4 | 1259.2 | 1343.9 | 1409.7 | 1464.4 | 1520.7 | 1616.2 | 1711.6 | |
| | | | | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | | | | |
| **Actual** | | | | | | | | | 02-Dec | | | | | | | | | |
| STJA 02624 | | | | | | | **51.0** | **54.3** | **75.4** | 65.2 | 22.7 | 22.7 | 18.3 | 29.2 | 29.6 | 13.0 | 13.0 | |
| | | | | | | | **54.2** | **65.1** | **58.3** | 66.5 | 22.7 | 22.7 | 18.3 | 29.2 | 29.6 | 13.0 | 13.0 | |
| | | | | | | | **53.4** | **44.6** | 65.2 | 44.3 | 22.7 | 22.7 | 12.7 | 29.2 | 29.6 | 13.0 | 13.0 | |
| | | | | | | | | | | | | | | | | | | |
| **Total** | **10.4** | **32.2** | **28.3** | **27.2** | **74.9** | **140.1** | **158.6** | **164.0** | **198.9** | **176.0** | **68.0** | **68.0** | **49.2** | **87.6** | **88.8** | **39.1** | **39.1** | |
| Total (cumulative) | 10.4 | 42.5 | 70.8 | 98.0 | 172.9 | 312.9 | 471.5 | 635.5 | 834.5 | 1010.5 | 1078.5 | 1146.5 | 1195.7 | 1283.3 | 1372.0 | 1411.1 | 1450.3 | |
| | | | | | | | | | | | | | | | | | | |
| STJA 01624 | **0.0** | **4.2** | **0.0** | **6.3** | **5.1** | **9.5** | **0.2** | **0.3** | **8.2** | | | | | | | | | |
| | 0.0 | 4.2 | 4.2 | 10.5 | 15.6 | 25.1 | 25.2 | 25.5 | 33.7 | 33.7 | 33.7 | 33.7 | 33.7 | 33.7 | 33.7 | 33.7 | 33.7 | |

*Thank you!*

UKRI Science and Technology Facilities Council ISIS

*PACE*

# Brille library and resolution

***brille*:**

Goals:

- irreducible Brillouin zone polyhedral for any space group
- Interpolation for any scalar, vector or matrix quantity for speed

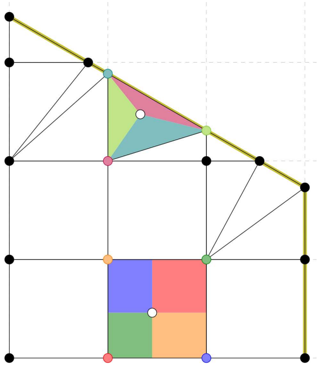
Features:

- C++/OpenMP with Python interface
- Interface:
  - Can drive modelling codes (as by Euphonic)
  - Can be driven by modelling codes (as by SpinW development version)

- Reduces up-front calculation by Fourier interpolation of force constant matricies (x 1/48 for simple cubic)

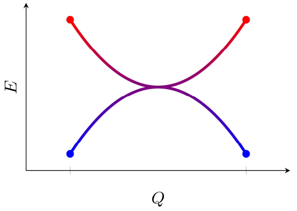
- Circumvents the interpolation problem across high symmetry directions/planes

`brille` implements a **hybrid grid/mesh**:



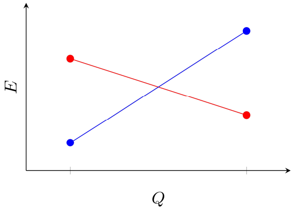
- ➢ Uses regular grid where possible
- ➢ Boundary cells use n-simplex mesh

degeneracy



eigenvector mixing prevents identification

accidental



distinct eigenvectors help identification

# Brille library and resolution

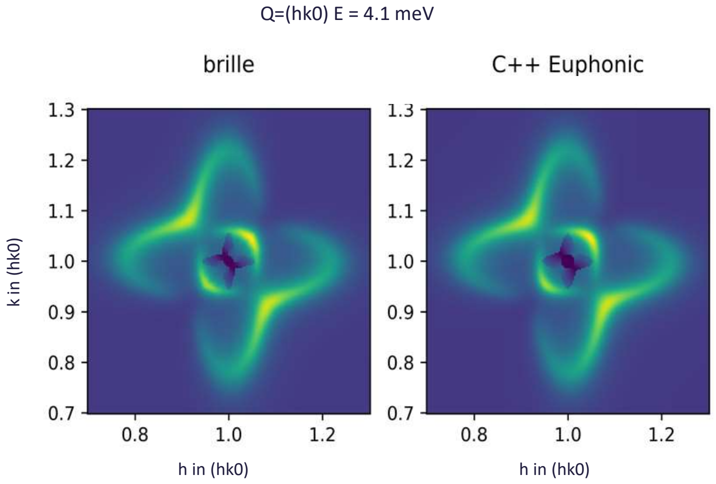
**Resolution convolution:**

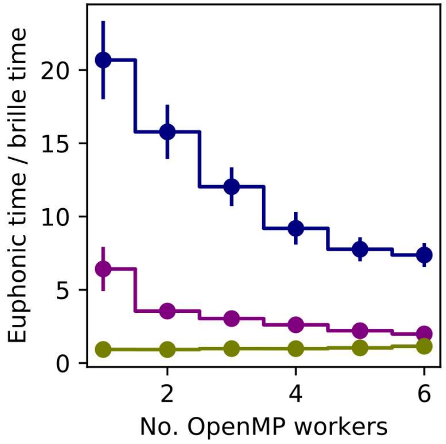
- Refactored existing Matlab instrument and detector classes
- Fast lookup for random sampling from instrument components

**Parameter optimisation:**

- ExtendsDesign calls for new classes to
  - wrap user-defined model functions with metadata
  - provide interface between models and parameter optimisers
- Planned support for models which take **Q**, (**Q**,E), `sqw`, or plotting axes as input
- Desire to support multiple parameter optimisation engines
  - local minimisers, e.g., Levenberg-Marquardt
  - global minimisers, e.g., NLOpt
  - derivative-free optimisers, e.g., DFO-LS
- Full details at https://github.com/pace-neutrons/pace-developers/blob/master/optimisation/design/Model_Optimisation_Design.md

niobium phonons





# PACE Goals

- Optimisation of HORACE through
  - Parallelization for multiple core computers and distributed computing, Data Analysis as a Service (DAaaS) and SCARF at STFC.
  - Handling of large datasets out of memory
- Parallelisation of TOBYFIT for the framework
- Computation of S(**Q**,$\omega$) from phonons from force constant matrix *(CASTEP, GULP, Phonopy)*
- Parallelisation of SpinW (spin wave modelling) – *now a separate project*
- Generic Application Programming Interface (API) to user function (Matlab, Python, compiled C++) and third party modelling codes
- Construction of a GUI based ‘workbench’ for managing analysis of data with refinement of parameters
- Mantid based manipulation and GUI based visualisation of powder data
- Handover as a product for operations at the project end

UKRI Science and Technology Facilities Council ISIS

*PACE*