MRC | Cognition and Brain Sciences Unit

# Parallel Computing in Matlab

**Tibor Auer**

MRC Cognition and Brain Sciences Unit, Methods group

# Structures applied

## SPM

- **Organised/”structured” storage of variables with any type**
  - SPM default settings (for fMRI):
    - *spm('Defaults','FMRI');*
    - *global defaults*
    - *defaults*
    - *defaults.realign.estimate (settings for motion correction estimation)*
      - *quality: 0.9000*
      - *interp: 2*
      - *wrap: [0 0 0]*
      - *sep: 4*
      - *fwhm: 5*
      - *rtm: 1*

# Structures applied

## SPM

- SPM job: matlabbatch[1]
  - *matlabbatch{1}.spm.spatial.realignunwarp.eoptions*
    - *quality: 0.9000*
    - *sep: 4*
    - *fwhm: 5*
    - *rtm: 0*
    - *einterp: 2*
    - *ewrap: [0 0 0]*
    - *weight: ''*

# Structures applied

## SPM

- SPM design: SPM =
  - *xY*: Data
  - *Sess*: Design settings (events)
  - *xX*: Design Matrix
  - *xCon*: Contrasts (only after specifying them)
  - xBF: [1x1 struct]: (canonical) hemodynamic base function
  - …

# Structures applied

## SPM

- SPM design: SPM =
  - *SPM.xBF:*
    - *UNITS:= 'secs'*
    - *Volterra: 1*
    - *Name: 'hrf'*
    - *length: 32.0625*
    - *order: 1*
    - *T: 32*
    - *T0: 17*
    - *dt: 0.0625*
    - *bf: [513x1 double]*
  - *SPM.xBF.UNITS = 'secs'*

# Structures applied

## Errors

- Error structure
  - Unhandled: *lasterror*
  - Handled: try-catch block (*test_error_call*)

  - What was the error…
    - *identifier: 'MATLAB:UndefinedFunction'*
    - *message: 'Undefined function or variable 'this_is_not_an_existing_function'.'*
    - *cause: {0x1 cell}*

  - **We have called an un-existing/undefined/un-found function…**

# Structures applied

## Errors

- Error structure (*test_error_call*)
  - … and where did it occurred (from bottom to top)
    - *stack: [2x1 struct]*
    - *E.stack(1)*
      - *file: 'D:\Workshop\2_MATLAB\test_error.m'*
      - *name: 'test_error'*
      - *line: 3*
    - *E.stack(2)*
      - *file: 'D:\Workshop\2_MATLAB\test_error_call.m'*
      - *name: 'test_error_call'*
      - *line: 1*
  - **… in line 3 of test_error called by line 1 of test_error_call**

# Structures applied

## qsub in MATLAB

- Wrapper: **qsubcellfun**

qsubcellfun(@function_name,{‘inp1_for_job1’ ‘inp1_for_job2’}, {‘inp2_for_job1’ ‘inp2_for_job2’})

E.g.: fprintf(format, data)

```
format = ‘%s’;
data = ‘a’;
fprintf(format,data)
```

# Structures applied

## qsub in MATLAB

- Wrapper: **qsubcellfun**

qsubcellfun(@function_name,{‘inp1_for_job1’ ‘inp1_for_job2’}, {‘inp2_for_job1’ ‘inp2_for_job2’})

E.g.: fprintf(format, data)

format = ‘%s’;
data = ‘a’;
fprintf(format,data)

format = {‘%s’ ‘%d’ ‘%f’}
data = {‘a’ ‘a’ ‘a’} or {‘a’ 10 3.1415}
qsubcellfun(@fprintf, format, data)

# Structures applied

## qsub

- Job folders: Queuing system results in Job*ID* folders

  - Job1_script.sh - qsub engine: last line shows you the qsub settings
  - matlab_metadata.mat - MATLAB Distributed Computing engine
  - Task1.common.mat - Text!: Date and time of access to the job
  - **Task1.diary.txt** **- Output to the Command window**
  - **Task1.in.mat** **- The actual job**
  - Task1.jobout.mat
  - **Task1.log** **- Output to the Terminal**
  - **Task1.out.mat** **- Results/Error of the job**
  - Task1.state.mat - Text!: job status

# Structures applied

## qsub

- Task1.in.mat
  - *createtime: 'Fri Dec 06 17:59:55 GMT 2013'*
  - *taskfunction: @fprintf* - Submitted function call
  - *argsin: {'%s' 'a'}* - Arguments of the function
  - *name: 'Task1'* - Name (qstat)
  - *nargout: 0*

- Actual command: *fprintf('%s', 'a')* - The actual job
  - argsin{1}: '%s'
  - Formatted print of string 'a'.

# Structures applied

## qsub

- Task1.out.mat - Results/Error of the job
  - *erroridentifier: ''*
  - *errormessage: ''*
  - *worker: [1x1 struct]*
  - *argsout: {}*
  - *errorstruct: []*
  - *finishtime: 'Fri Dec 06 18:00:09 GMT 2013'*

  - *worker:*
    - *Host: 'node-d08.mrc-cbu.cam.ac.uk'* - The actual machine
    - *ComputerType: 'GLNXA64'*
    - *ProcessId: 31540* - ID (qstat)