The University Of Sheffield.

Writing Object Oriented MATLAB For Parallel Compute: Challenges and Successes

T.J. Rogers

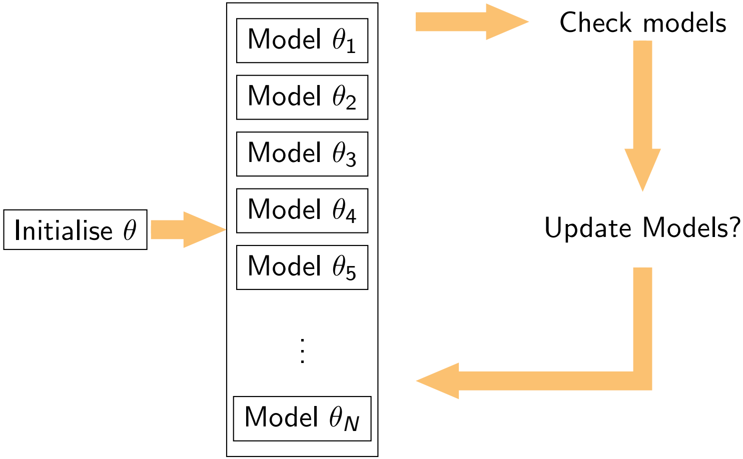
October, 2021

WHY OBJECT ORIENTED?

- This is not Matlab specific...
- Any time we have a number of repeated components with their own properties (encapsulation)
- The aim is to avoid repeated effort and increase readability
- We can hide away internal intermediate computational steps that we don't need the user to interact with (abstraction)
- Let's see a scientific/engineering example

SMC IN ONE MINUTE!

The aim is to infer a posterior distribution of some parameters θ



Initialise θ

Model θ_1

Model θ_2

Model θ_3

Model θ_4

Model θ_5

$\vdots$

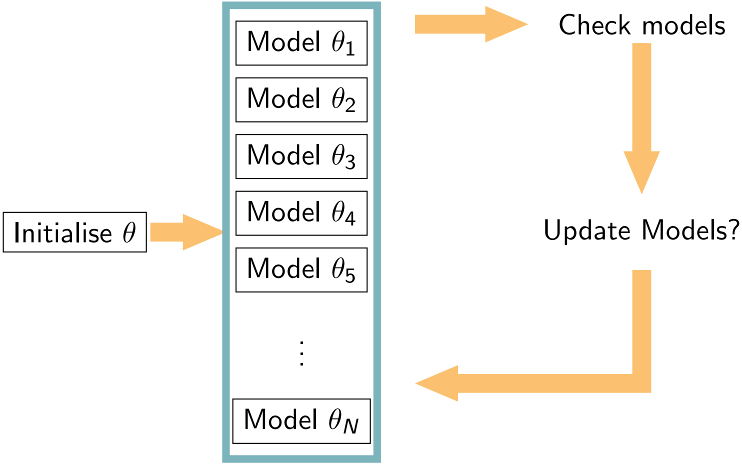
Model θ_N

Check models

Update Models?

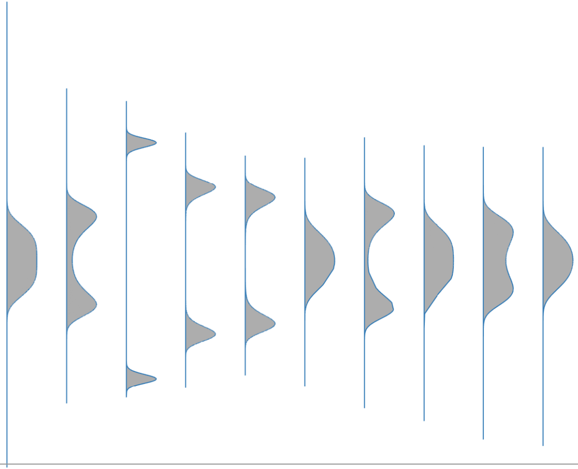
SMC IN ONE MINUTE!

The aim is to infer a posterior distribution of some parameters θ

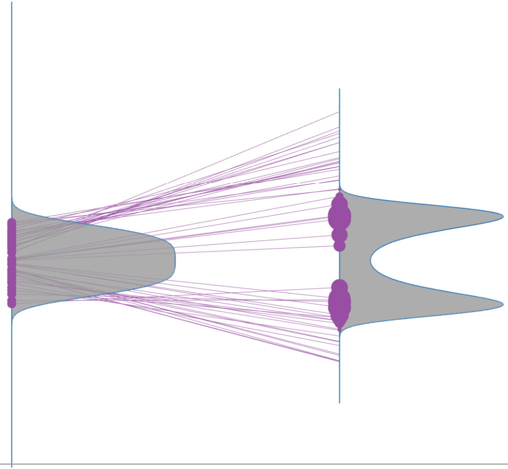


PARTICLE FILTERING
A very brief introduction

Estimating sequences of probability distributions:



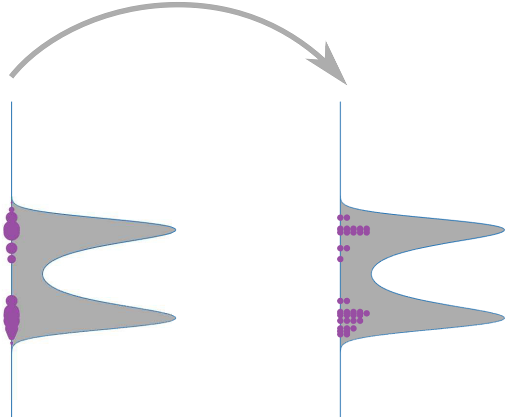
Particle Propagation and Weighting:



PARTICLE FILTERING

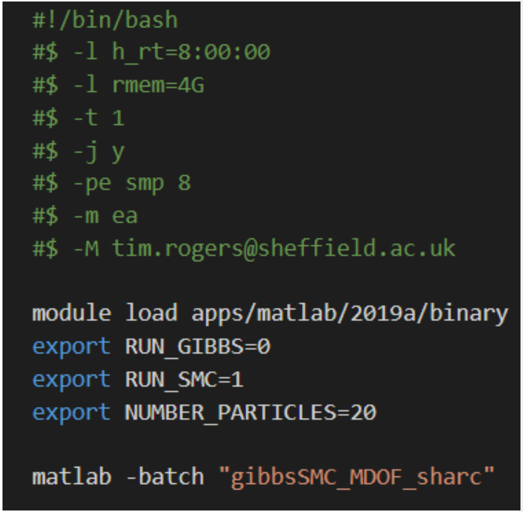
A very brief introduction

Resampling:

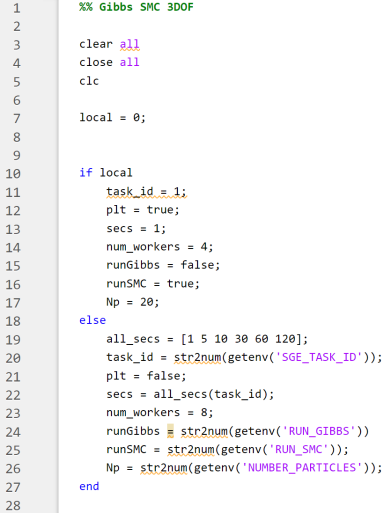


HOW IN MATLAB?

Setup

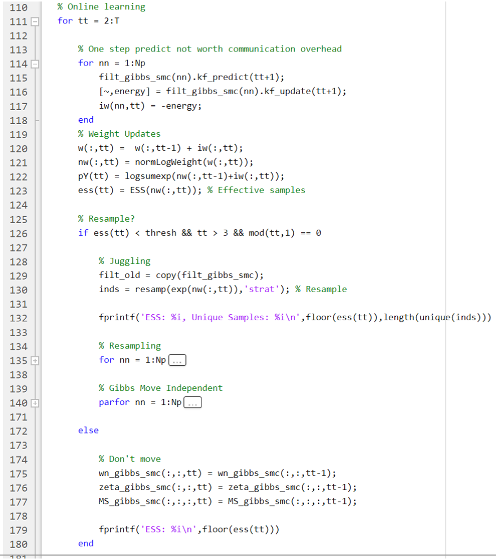


```bash
#!/bin/bash
#$ -l h_rt=8:00:00
#$ -l rmem=4G
#$ -t 1
#$ -j y
#$ -pe smp 8
#$ -m ea
#$ -M tim.rogers@sheffield.ac.uk

module load apps/matlab/2019a/binary
export RUN_GIBBS=0
export RUN_SMC=1
export NUMBER_PARTICLES=20

matlab -batch "gibbsSMC_MDOF_sharc"
```



```matlab
%% Gibbs SMC 3DOF

clear all
close all
clc

local = 0;


if local
    task_id = 1;
    plt = true;
    secs = 1;
    num_workers = 4;
    runGibbs = false;
    runSMC = true;
    Np = 20;
else
    all_secs = [1 5 10 30 60 120];
    task_id = str2num(getenv('SGE_TASK_ID'));
    plt = false;
    secs = all_secs(task_id);
    num_workers = 8;
    runGibbs = str2num(getenv('RUN_GIBBS'))
    runSMC = str2num(getenv('RUN_SMC'));
    Np = str2num(getenv('NUMBER_PARTICLES'));
end

```

HOW IN MATLAB?
Full Algorithm



```
% Online learning
for tt = 2:T

    % One step predict not worth communication overhead
    for nn = 1:Np
        filt_gibbs_smc(nn).kf_predict(tt+1);
        [~,energy] = filt_gibbs_smc(nn).kf_update(tt+1);
        iw(nn,tt) = -energy;
    end
    % Weight Updates
    w(:,tt) =  w(:,tt-1) + iw(:,tt);
    nw(:,tt) = normLogWeight(w(:,tt));
    pY(tt) = logsumexp(nw(:,tt-1)+iw(:,tt));
    ess(tt) = ESS(nw(:,tt)); % Effective samples

    % Resample?
    if ess(tt) < thresh && tt > 3 && mod(tt,1) == 0

        % Juggling
        filt_old = copy(filt_gibbs_smc);
        inds = resamp(exp(nw(:,tt)),'strat'); % Resample

        fprintf('ESS: %i, Unique Samples: %i\n',floor(ess(tt)),length(unique(inds)))

        % Resampling
        for nn = 1:Np...

        % Gibbs Move Independent
        parfor nn = 1:Np...

    else

        % Don't move
        wn_gibbs_smc(:,:,tt) = wn_gibbs_smc(:,:,tt-1);
        zeta_gibbs_smc(:,:,tt) = zeta_gibbs_smc(:,:,tt-1);
        MS_gibbs_smc(:,:,:,tt) = MS_gibbs_smc(:,:,:,tt-1);

        fprintf('ESS: %i\n',floor(ess(tt)))
    end
```

One step ahead – check model quality

Update weights and diagnostics

Need to move?

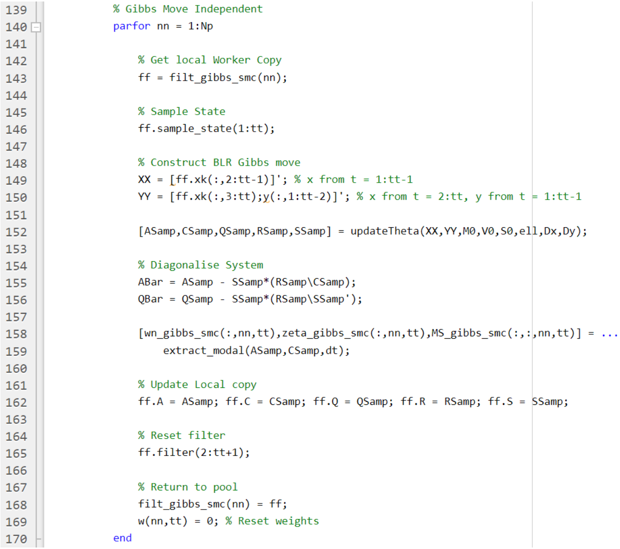
Resampling

Move each model with Gibbs in parallel

No move – copy old parameters

HOW IN MATLAB?

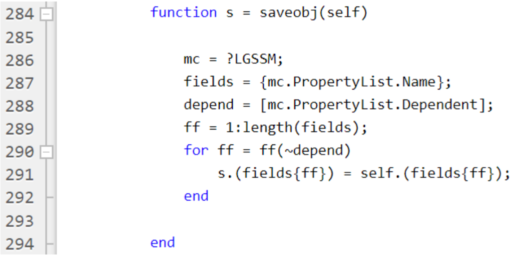
The Tricky Bit



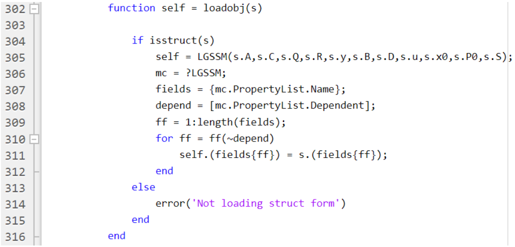
```
        % Gibbs Move Independent
        parfor nn = 1:Np

            % Get local Worker Copy
            ff = filt_gibbs_smc(nn);

            % Sample State
            ff.sample_state(1:tt);

            % Construct BLR Gibbs move
            XX = [ff.xk(:,2:tt-1)]'; % x from t = 1:tt-1
            YY = [ff.xk(:,3:tt);y(:,1:tt-2)]'; % x from t = 2:tt, y from t = 1:tt-1

            [ASamp,CSamp,QSamp,RSamp,SSamp] = updateTheta(XX,YY,M0,V0,S0,ell,Dx,Dy);

            % Diagonalise System
            ABar = ASamp - SSamp*(RSamp\CSamp);
            QBar = QSamp - SSamp*(RSamp\SSamp');

            [wn_gibbs_smc(:,nn,tt),zeta_gibbs_smc(:,nn,tt),MS_gibbs_smc(:,:,nn,tt)] = ...
                extract_modal(ASamp,CSamp,dt);

            % Update Local copy
            ff.A = ASamp; ff.C = CSamp; ff.Q = QSamp; ff.R = RSamp; ff.S = SSamp;

            % Reset filter
            ff.filter(2:tt+1);

            % Return to pool
            filt_gibbs_smc(nn) = ff;
            w(nn,tt) = 0; % Reset weights
        end
```

HOW IN MATLAB?

The Tricky Bit II



```
function s = saveobj(self)


    mc = ?LGSSM;
    fields = {mc.PropertyList.Name};
    depend = [mc.PropertyList.Dependent];
    ff = 1:length(fields);
    for ff = ff(~depend)
        s.(fields{ff}) = self.(fields{ff});
    end

end
```



```
function self = loadobj(s)

    if isstruct(s)
        self = LGSSM(s.A,s.C,s.Q,s.R,s.y,s.B,s.D,s.u,s.x0,s.P0,s.S);
        mc = ?LGSSM;
        fields = {mc.PropertyList.Name};
        depend = [mc.PropertyList.Dependent];
        ff = 1:length(fields);
        for ff = ff(~depend)
            self.(fields{ff}) = s.(fields{ff});
        end
    else
        error('Not loading struct form')
    end
end
```

The
University
Of
Sheffield.

Writing Object Oriented MATLAB For Parallel Compute: Challenges and Successes

T.J. Rogers

October, 2021