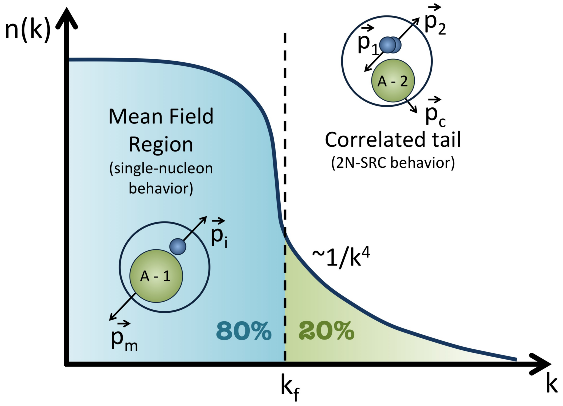
Correlated Fermi Gas

Afroditi Papadopoulou
On behalf of the MIT-Nuclear Group

February 5, 2019

Electron Experiments

~20% of nucleons form
short range correlated (SRC) pairs



arXiv:1710.07966

Objective

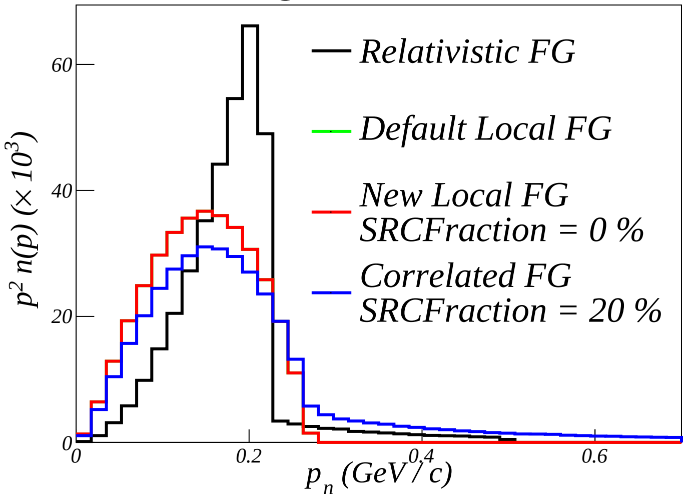
- Implementation of Correlated Fermi Gas Model

Important

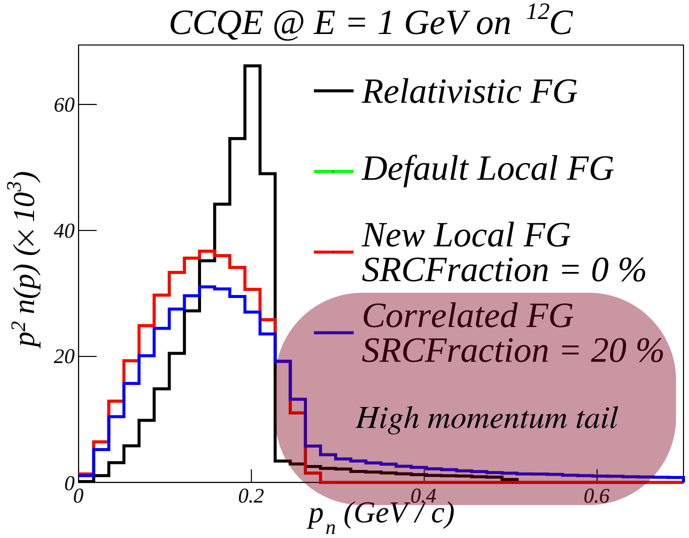
- Free parameter: SRCFraction
- Reproduce Local Fermi Gas when SRCFraction = 0

Correlated Fermi Gas

CCQE @ E = 1 GeV on ^{12}C



Correlated Fermi Gas



Modified Files

- config/LocalFGM.xml
- src/Physics/NuclearState/LocalFGM.cxx
- src/Physics/NuclearState/LocalFGM.h

Github

Repository

https://github.com/afropapp13/Generator.git

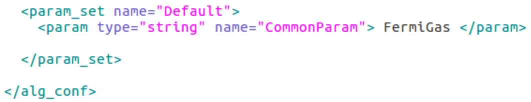
Branch

devel_cfg

Code Modifications

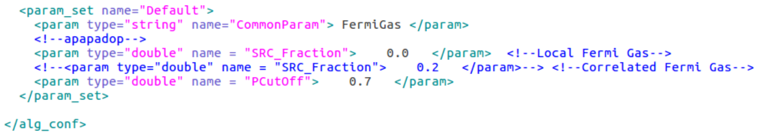
LocalFG.xml

Default Code



```
  <param_set name="Default">
    <param type="string" name="CommonParam"> FermiGas </param>

  </param_set>

</alg_conf>
```

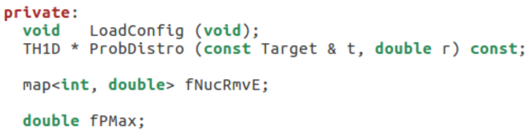
Modified Code



```
  <param_set name="Default">
    <param type="string" name="CommonParam"> FermiGas </param>
    <!--apapadop-->
    <param type="double" name = "SRC_Fraction">    0.0   </param>  <!--Local Fermi Gas-->
    <!--<param type="double" name = "SRC_Fraction">    0.2   </param>--> <!--Correlated Fermi Gas-->
    <param type="double" name = "PCutOff">    0.7   </param>
  </param_set>

</alg_conf>
```

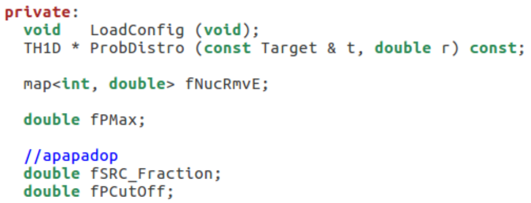
LocalFG.h

Default Code



```
private:
  void   LoadConfig (void);
  TH1D * ProbDistro (const Target & t, double r) const;

  map<int, double> fNucRmvE;

  double fPMax;
```

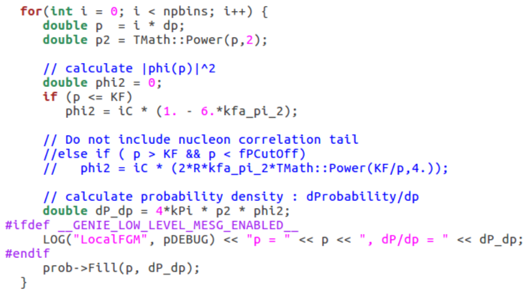
Modified Code



```
private:
  void   LoadConfig (void);
  TH1D * ProbDistro (const Target & t, double r) const;

  map<int, double> fNucRmvE;

  double fPMax;

  //apapadop
  double fSRC_Fraction;
  double fPCutOff;
```

LocalFG.cxx

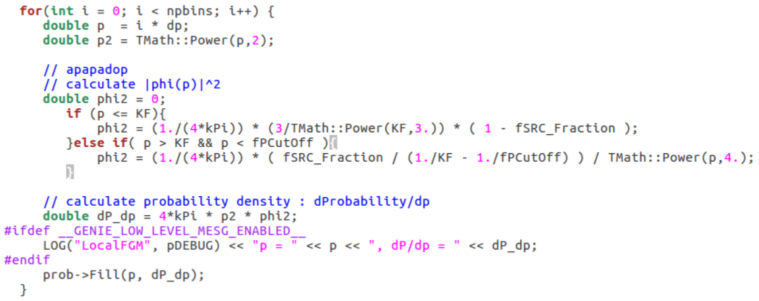
Default Code



```
  for(int i = 0; i < npbins; i++) {
     double p  = i * dp;
     double p2 = TMath::Power(p,2);

     // calculate |phi(p)|^2
     double phi2 = 0;
     if (p <= KF)
        phi2 = iC * (1. - 6.*kfa_pi_2);

     // Do not include nucleon correlation tail
     //else if ( p > KF && p < fPCutOff)
     //   phi2 = iC * (2*R*kfa_pi_2*TMath::Power(KF/p,4.));

     // calculate probability density : dProbability/dp
     double dP_dp = 4*kPi * p2 * phi2;
#ifdef __GENIE_LOW_LEVEL_MESG_ENABLED__
     LOG("LocalFGM", pDEBUG) << "p = " << p << ", dP/dp = " << dP_dp;
#endif
     prob->Fill(p, dP_dp);
  }
```

LocalFG.cxx

Modified Code



```
  for(int i = 0; i < npbins; i++) {
     double p  = i * dp;
     double p2 = TMath::Power(p,2);

     // apapadop
     // calculate |phi(p)|^2
     double phi2 = 0;
        if (p <= KF){
           phi2 = (1./(4*kPi)) * (3/TMath::Power(KF,3.)) * ( 1 - fSRC_Fraction );
        }else if( p > KF && p < fPCutOff ){
           phi2 = (1./(4*kPi)) * ( fSRC_Fraction / (1./KF - 1./fPCutOff) ) / TMath::Power(p,4.);
        }

     // calculate probability density : dProbability/dp
     double dP_dp = 4*kPi * p2 * phi2;
#ifdef __GENIE_LOW_LEVEL_MESG_ENABLED__
     LOG("LocalFGM", pDEBUG) << "p = " << p << ", dP/dp = " << dP_dp;
#endif
     prob->Fill(p, dP_dp);
  }
```

GENIE-Doc-36-v2

Thank you!

Questions ?

Backup Slides