**主变风险预测及基于主变状态的电网实时风险分析接入规范**

**V1.0**

**西安交通大学**

目 录

[1. 接口说明 3](#_Toc515958349)

[1.1 变压器故障后电网潮流快速估算及实时风险分析（直流潮流法） 4](#_Toc515958350)

[1.2 变压器故障后电网潮流快速估算及实时风险分析（PQ分解法） 5](#_Toc515958351)

[1.3 变压器故障后负荷削减及运行优化策略 6](#_Toc515958352)

[1.4 变压器风险评估 7](#_Toc515958353)

[2. 输入参数说明 8](#_Toc515958354)

[2.1电网风险评估直流潮流法 8](#_Toc515958355)

[2.2电网风险评估PQ分解法 9](#_Toc515958356)

[2.3电网负荷削减及运行优化策略算法 10](#_Toc515958357)

[2.4 变压器风险评估 11](#_Toc515958358)

[3. 输出参数说明 11](#_Toc515958359)

[3.1电网风险评估直流潮流法 11](#_Toc515958360)

[3.2 电网风险评估PQ法结果 12](#_Toc515958361)

[3.3电网风险评估负荷消减结果 12](#_Toc515958362)

[3.4 变压器风险评估结果 12](#_Toc515958363)

# 接口说明

所有算法采用统一调用方式：

UniParameter uniParameter = new UniParameter();

//TODO 设置uniParameter  
UniSolver uniSolver = new UniSolver();  
UniResult uniResult = uniSolver.solve(uniParameter, "COOL");

统一输入参数对象：

public class UniParameter {  
 private IFhnlpgBase iFhnlpgBase;  
 private IFhnlpgInitial iFhnlpgInitial;  
 private IFhnlpgOnLoad iFhnlpgOnLoad;  
 private IFhnlpgTRise iFhnlpgTRise;  
 private IFhnlpgResistance iFhnlpgResistance;  
 private IFhnlpgRatio iFhnlpgRatio;  
 private IFhnlpgLimit iFhnlpgLimit;  
 private IFhnlpgCool iFhnlpgCool;  
 private IFhnlpgTStatus iFhnlpgTStatus;  
 private IDwfxpgDW iDwfxpgDW;  
 private IDwfxpgPQ iDwfxpgPQ;

private IFXPG iFxpg;

}

统一输出对象：

public class UniResult {  
 public int errcode = 0; //0：成功；-1:失败  
 public String errmsg; //失败原因  
 public String mode; //算法缩写：HST,ZCFHNLPG等  
 public OCool oCool;  
 public OHst oHst;  
 public OZcfhnlpg oZcfhnlpg;  
 public OCqfhnlpg oCqfhnlpg;  
 public ODqfhnlpg oDqfhnlpg;  
 public OFhzt oFhzt;  
 public OZLCL oZlcl;  
 public OFHXJ oFhxj;  
 public OPQ oPq;

public OFXPG oFxpg;

}

## 1.1 变压器故障后电网潮流快速估算及实时风险分析（直流潮流法）

**1）算法模型**

|  |  |
| --- | --- |
| 算法名称 | 变压器故障后电网潮流快速估算及实时风险分析算法 |
| 输入数据 | 基本要求：   1. 某变压器故障后的电网数据   节点标识、电压、功率，线路电阻、线路电抗、线路对地导纳、线路容量，变压器标准变比、变压器容量 |
| 输出数据 | 1. 电网潮流分布及实时风险结果 |

**2）算法接口**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 接口名称 | 类名 | UniSolver | | |
| 接口名 | Solve（） | | |
| 接口原型 | | UniResultuniResult = uniSolver.solve(uniParameter, "ZLCL"); | | |
| 参数列表 | | 说明 | 参数类型 | 备注 |
| iDwfxpgDW | | 电网参数 | 对象 | [详见2.](#_设备基本信息_EquipmentData)1 |
| 返回值 | | 说明 | 参数类型 | 备注 |
| oZlcl | | 电网潮流及电网风险 | 对象 | [详见3.1](#_设备基本信息_EquipmentData) |

## 1.2 变压器故障后电网潮流快速估算及实时风险分析（PQ分解法）

**1）算法模型**

|  |  |
| --- | --- |
| 算法名称 | 变压器故障后电网潮流快速估算及实时风险分析算法 |
| 输入数据 | 基本要求：   1. 某变压器故障后的电网数据   节点标识、电压、功率，线路电阻、线路电抗、线路对地导纳、线路容量，变压器标准变比、变压器容量 |
| 输出数据 | 1. 电网潮流分布及实时风险结果 |

**2）算法接口**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 接口名称 | 类名 | UniSolver | | |
| 接口名 | Solve（） | | |
| 接口原型 | | UniResultuniResult = uniSolver.solve(uniParameter, "PQ"); | | |
| 参数列表 | | 说明 | 参数类型 | 备注 |
| iDwfxpgPQ | |  | 对象 | 详见2.2 |
| 返回值 | | 说明 | 参数类型 | 备注 |
| oPq | | PQ分解法结果 |  | [详见3.2](#_设备基本信息_EquipmentData) |

## 1.3 变压器故障后负荷削减及运行优化策略

**1）算法模型**

|  |  |
| --- | --- |
| 算法名称 | 变压器故障后负荷削减计算及运行优化措施 |
| 输入数据 | 1. 某变压器故障后的电网数据   节点标识、节点负荷功率，节点发电功率、支路起始节点、终止节点支路标识、支路容量 |
| 输出数据 | 1. 变压器故障后负荷削减结果及运行优化措施 |

**2）算法接口**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 接口名称 | 类名 | UniSolver | | |
| 接口名 | Solve（） | | |
| 接口原型 | | UniResultuniResult = uniSolver.solve(uniParameter, "FHXJ"); | | |
| 参数列表 | | 说明 | 参数类型 | 备注 |
|  | | 负荷消减法输入完全同直流潮流法 |  | 参见1.1 |
| 返回值 | | 说明 | 参数类型 | 备注 |
| oFhxj | | 负荷削减量及优化建议 |  | 详见3.3 |

## 1.4 变压器风险评估

**1）算法模型**

|  |  |
| --- | --- |
| 算法名称 |  |
| 输入数据 |  |
| 输出数据 |  |

**2）算法接口**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 接口名称 | 类名 | UniSolver | | |
| 接口名 | Solve（） | | |
| 接口原型 | | UniResultuniResult = uniSolver.solve(uniParameter, "FXPG"); | | |
| 参数列表 | | 说明 | 参数类型 | 备注 |
| iFxpg | | 风险评估输入 | 对象 |  |
| 返回值 | | 说明 | 参数类型 | 备注 |
| oFxpg | | 负荷削减量及优化建议 |  | 详见3.4 |

# 输入参数说明

## 2.1电网风险评估直流潮流法(IDwfxpgDW)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **构造函数** | public IDwfxpgDW(ArrayList<CBus> bus, ArrayList<CLoad> load, ArrayList<CGenerator> generator, ArrayList<CLine> line) {  Bus = bus;  Load = load;  Generator = generator;  Line = line; } | | | |
| **参数名称** | **代码值** | **必填** | **默认值** | **备注** |
| 电网节点 | Bus | Y |  |  |
| 电网中有功负荷 | Load | Y |  |  |
| 电网中有功发电量 | Generator | Y |  |  |
| 电网支路数据 | Line | Y |  |  |
| **构造函数** | public CBus(int id, int type) {  this.id = id;  this.type = type; } | | | |
| **参数名称** | **代码值** | **必填** | **默认值** | **备注** |
| 节点标识 | **id** | Y |  | 编号必须从1开始，依次递增 |
| 节点类型 | type | Y |  | PQ节点为1，PV节点为2，平衡节点最后输入为3 |
| **构造函数** | public CLoad(int id, double pl) {  this.id = id;  this.pl = pl; } | | | |
| **参数名称** | **代码值** | **必填** | **默认值** | **备注** |
| 节点标识 | **id** | Y |  |  |
| 节点有功负荷 | **pl** | Y |  | 标幺值 |
| **构造函数** | public CGenerator(int id, double pg) {  this.id = id;  this.pg = pg; } | | | |
| **参数名称** | **代码值** | **必填** | **默认值** | **备注** |
| 节点标识 | **id** | Y |  |  |
| 节点有功发电量 | **pg** | Y |  | 标幺值 |
| **构造函数** | public CLine(int sid, int eid, int sn, double resistance, double capacity) {  this.sid = sid;  this.eid = eid;  this.sn = sn;  this.resistance = resistance;  this.capacity = capacity; } | | | |
| **参数名称** | **代码值** | **必填** | **默认值** | **备注** |
| 支路起始节点标识 | **sid** | Y |  | 对应bus的id |
| 支路终点节点标识 | **eid** | Y |  |  |
| 支路标识 | **sn** | Y |  | 线路为1，变压器支路为0 |
| 支路电抗 | **resistance** | Y |  | 标幺值 |
| 支路容量 | capacity | Y |  | 标幺值 |

## 2.2电网风险评估PQ分解法

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **构造函数** | public IDwfxpgPQ(ArrayList<CBus> bus, ArrayList<CLine> line) {  Bus = bus;  Line = line; } | | | |
| **参数名称** | **代码值** | **必填** | **默认值** | **备注** |
| 电网节点数量 | Bus | Y |  |  |
| 电网支路（线路）数 | Line | Y |  |  |
| **构造函数** | public CBus(int id, int type, double U, double a, double P, double Q) {  this.id = id;  this.type = type;  this.U = U;  this.a = a;  this.P = P;  this.Q = Q; } | | | |
| **参数名称** | **代码值** | **必填** | **默认值** | **备注** |
| 节点标识 | **Id** | Y |  |  |
| 节点类型 | Type | Y |  | PQ为1，PV为2，平衡节点为3 |
| 节点电压 | U | Y |  | 标幺值 |
| 节点相角 | a | Y |  | 标幺值 |
| 节点有功功率 | P | Y |  | 负荷为负，发电量为正，标幺值 |
| 节点无功功率 | Q | Y |  | 容性为正，感性为负，标幺值 |
| **构造函数** | public CLine(int sid, int eid, double Rij, double Xij, double B0, double RT, double XT, double KT, double W, double capacity) {  this.sid = sid;  this.eid = eid;  this.Rij = Rij;  this.Xij = Xij;  this.B0 = B0;  this.RT = RT;  this.XT = XT;  this.KT = KT;  this.W = W;  this.capacity = capacity; } | | | |
| **参数名称** | **代码值** | **必填** | **默认值** | **备注** |
| 支路起始节点标识 | **Sid** | Y |  |  |
| 支路终点节点标识 | **Eid** | Y |  |  |
| 线路电阻 | **Rij** | Y |  | 标幺值，变压器支路此项为0 |
| 电抗 | **Xij** | Y |  | 标幺值，变压器支路此项为0 |
| 支路对地导纳 | **BO** | Y |  | 标幺值 |
| 变压器支路电阻 | **RT** | Y |  | 标幺值，若为线路支路此项为0 |
| 变压器支路电抗 | **XT** | Y |  | 标幺值，若为线路支路此项为0 |
| 变压器标准变比 | **KT** | Y |  | 标幺值 |
| 折算标志 | **W** | Y |  | 变压器等效支路阻抗折算高压侧为0，低压侧为1 |
| 支路容量 | **capacity** | Y |  | 标幺值 |

## 2.3电网负荷削减及运行优化策略算法

## 2.4 变压器风险评估

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **构造函数** | public class IFXPG {  private int score = 80; //健康评分  private String scoreMsg = ""; //状态评价的风险因素描述  private String diagnoseResult = ""; //诊断结果：异常状态描述  private double t = 10; //运行年限（年）  private double V = 180; //变压器容量（MVA）  private int x = 0; //设备标识  private double I = 800; //高压侧电流  }  public IFXPG(int score, double t, double V, int x, double I) {  this.score = score;  this.t = t;  this.V = V;  this.x = x;  this.I = I; }  public void setScoreMsg(String scoreMsg) {  this.scoreMsg = scoreMsg; }  public void setDiagnoseResult(String diagnoseResult) {  this.diagnoseResult = diagnoseResult; } | | | |
| **参数名称** | **代码值** | **必填** | **默认值** | **备注** |
| 健康评分 | Score | Y |  |  |
| 运行年限（年） | T | Y |  |  |
| 变压器容量（MVA） | V | Y |  |  |
| 设备标识 | X | Y |  |  |
| 高压侧电流 | I | Y |  |  |
| 状态评价的风险因素描述 | scoreMsg |  |  |  |
| 诊断结果：异常状态描述 | diagnoseResult |  |  |  |

# 输出参数说明

## 3.1电网风险评估直流潮流法

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **类** | public class OZLCL {  public ArrayList<CLinePIJ> PIJ;  public ArrayList<LineMsg> lineMsgs;  public ArrayList<BusMsg> busMsgs;  } | | | | |
| **属性名称** | **代码值** | | | **备注** | |
| 线路潮流 | PIJ | | |  | |
| 线路风险 | lineMsgs | | |  | |
| 节点相角 | busMsgs | | |  | |
| **构造函数** | public CLinePIJ(int sid, int eid, double pij) {  this.sid = sid;  this.eid = eid;  this.pij = pij; } | | | | | |
| **参数名称** | **代码值** | **必填** | **默认值** | | **备注** | |
|  |  |  |  | |  | |
|  |  |  |  | |  | |
|  |  |  |  | |  | |
|  |  |  |  | |  | |
|  |  |  |  | |  | |
|  |  |  |  | |  | |
|  |  |  |  | |  | |
|  |  |  |  | |  | |
|  |  |  |  | |  | |
| **构造函数** | public LineMsg(int sid, int eid, String msg1, int status) {  this.sid = sid;  this.eid = eid;  this.msg1 = msg1;  this.status = status; } | | | | | |
| **参数名称** | **代码值** | **必填** | **默认值** | | **备注** | |
|  |  |  |  | |  | |
|  |  |  |  | |  | |
|  |  |  |  | |  | |
|  |  |  |  | |  | |
|  |  |  |  | |  | |
|  |  |  |  | |  | |
|  |  |  |  | |  | |
|  |  |  |  | |  | |
|  |  |  |  | |  | |
| **构造函数** | public BusMsg(int id, String msg1, int status) {  this.id = id;  this.msg1 = msg1;  this.status = status; } | | | | | |
| **参数名称** | **代码值** | **必填** | **默认值** | | **备注** | |
|  |  |  |  | |  | |
|  |  |  |  | |  | |
|  |  |  |  | |  | |
|  |  |  |  | |  | |
|  |  |  |  | |  | |
|  |  |  |  | |  | |
|  |  |  |  | |  | |
|  |  |  |  | |  | |
|  |  |  |  | |  | |

## 3.2 电网风险评估PQ法结果

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **类** | public class OPQ {  public Vector U; //节点电压  public Vector a; //节点相角  public Vector P; //节点有功  public Vector Q; //节点无功  public Vector I; //起始节点  public Vector J; //终止节点  public Complex Sph;  public VectorComplex Sij;  public VectorComplex Sji;  public VectorComplex deltaSij;  public VectorComplex S;  public Complex sumdeltaS;  public int k; //迭代次数  public ArrayList<LineMsg> lineMsgs;  public ArrayList<BusMsg> busMsgs;  } | | | | |
| **属性名称** | **代码值** | | | **备注** | |
| 迭代次数 |  | | | 若迭代次数k大于20,只输出“PQ分解法无法收敛，是否使用线性潮流估算方法的直流法” | |
| 平衡节点标识 |  | | |  | |
| 平衡节点复功率 |  | | | 标幺值 | |
| 节点电压 |  | | | 标幺值 | |
| 节点相角 |  | | | 标幺值 | |
| 各节点复功率 |  | | | 标幺值 | |
| 各节点有功功率 |  | | | 标幺值 | |
| 各节点无功功率 |  | | | 标幺值 | |
| 各支路功率 |  | | | 标幺值 | |
| 电网总损耗 |  | | | 标幺值 | |
| 各支路功率损耗 |  | | | 标幺值 | |
| 重载支路负载率 |  | | | 是否有越限风险 | |
| 节点电压情况 |  | | | 是否有越限风险 | |
| **构造函数** | public Complex(double a, double b) {  this.a = a;  this.b = b; } | | | | | |
| **参数名称** | **代码值** | **必填** | **默认值** | | **备注** | |
|  |  |  |  | |  | |
|  |  |  |  | |  | |
|  |  |  |  | |  | |
|  |  |  |  | |  | |
|  |  |  |  | |  | |
|  |  |  |  | |  | |
|  |  |  |  | |  | |
| **构造函数** | public class VectorComplex {  private Complex[] c;  private int size;  } | | | | | |
| **参数名称** | **代码值** | **必填** | **默认值** | | **备注** | |
|  |  |  |  | |  | |
|  |  |  |  | |  | |
|  |  |  |  | |  | |
|  |  |  |  | |  | |
|  |  |  |  | |  | |
|  |  |  |  | |  | |
|  |  |  |  | |  | |

## 3.3电网风险评估负荷消减结果

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **构造函数** | public class OFHXJ extends OZLCL {  public int status;  public String msg1;  public double f;  } | | | |
| **参数名称** | **代码值** | **必填** | **默认值** | **备注** |
| 负荷消减结论 | msg1 | Y |  |  |
| 负荷量 | f | Y |  |  |
| 状态标志 | status | Y |  | 当status=0时，父对象OZLCL没有数据 |

## 3.4 变压器风险评估结果

|  |  |  |
| --- | --- | --- |
| **类** | public class OFXPG {  public String riskLevel;  public String riskDisp;  public String scoreMsg; //状态评价的风险因素描述  public String diagnoseResult; //诊断结果：异常状态描述 } | |
| **属性名称** | **代码值** | **备注** |
| 风险等级 | FXDJ |  |
| 风险来源 | FXYS |  |