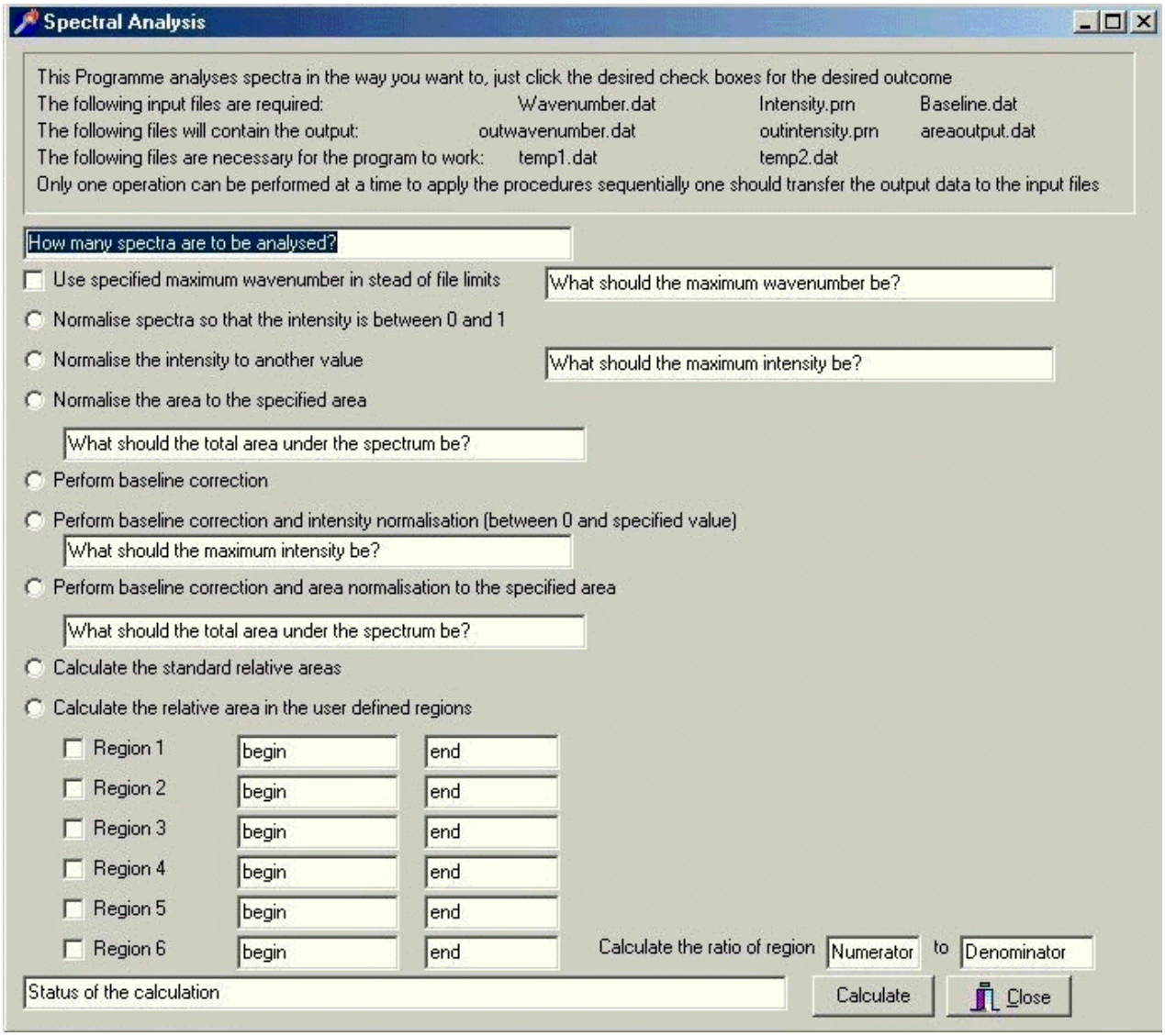
APPENDIX

Aspects of solid-state chemistry of fly ash and ultramarine pigments

A.PROGRAMS

A1. SPECTRAL MANIPULATIONS PROGRAM




```
unit Spectra_an;
{This program incorporates previous programs and allows the user to
manipulate spectra in several ways and calculate relative areas as defined by the user.}
interface
{The area is calculated by the trapezoidal Rule as described in
    R Ellis, D Gulick, Calculus With Analytical Geometry, 5th edition, Saunders College
    Publishing, Fort Worth, 1994, pp 470-471}
uses
  Windows, Messages, SysUtils, Classes, Graphics, Controls, Forms, Dialogs,
  StdCtrls, Buttons, ExtCtrls, math;
type
   TForm1 = class(TForm)
   CloseButton: TBitBtn;
   Bevel1: TBevel;
   intro_Label: TLabel;
   infiles_Label: TLabel;
   wavenumber_Label: TLabel;
   intensity_Label: TLabel;
   baseline_Label: TLabel;
   outfiles_Label: TLabel;
   outwavenumber_Label: TLabel;
```

```
    outintensity_Label: TLabel;
    added_programmes_Label: TLabel;
    temp1_Label: TLabel;
    temp2_Label: TLabel;
    sequential_Label: TLabel;
    CalculateButton: TButton;
    endbox: TEdit;
    areaout_Label: TLabel;
    Norm_int_radio: TRadioButton;
    numberofspectrainput: TEdit;
    Normarea_radio: TRadioButton;
    Norm_area_input: TEdit;
    Base_corr_radio: TRadioButton;
    base_int_corr_radio: TRadioButton;
    base_area_corr_radio: TRadioButton;
    norm_area_input2: TEdit;
    standard_area_calc_radio: TRadioButton;
    costum_area_calc_radio: TRadioButton;
    region1_CheckBox: TCheckBox;
    region1begin: TEdit;
    region1end: TEdit;
    region2_CheckBox: TCheckBox;
    region2begin: TEdit;
    region2end: TEdit;
    region3_CheckBox: TCheckBox;
    region3begin: TEdit;
    region3end: TEdit;
    region4_CheckBox: TCheckBox;
    region4begin: TEdit;
    region4end: TEdit;
    region5_CheckBox: TCheckBox;
    region5begin: TEdit;
    region5end: TEdit;
    region6_CheckBox: TCheckBox;
    region6begin: TEdit;
    region6end: TEdit;
    calc_ratio_Label: TLabel;
    numeratoregion: TEdit;
    ratio_to_Label: TLabel;
    ratiodenominator: TEdit;
    MaximumvalueCheckBox: TCheckBox;
    maxwaveinput: TEdit;
    othermaxint_radio: TRadioButton;
    input_maxint: TEdit;
    input_maxint2: TEdit;
    procedure CalculateButtonClick(Sender:
TObject);
  end;
var
  Form1: TForm1;
  datasize, topratioarea, bottomratioarea: integer;
  maximumwavelength, maxintensity: real;
  total, normarea, topratioareareal, bottomratioareareal: real;
  intensityarray, basearray, areaarray: array of array of real;
  totalint, flag, areaarraydimension: integer;
  I, J, A, B, err, temp: integer;
  interval: array[0..11]of integer;
  wavearray, minarray, maxarray: array of real;
  namearray: array of string;
  textfile, textfile2: text;
  region1min, region1max, region2min, region2max: real;
  region3min, region3max, region4min, region4max: real;
  region5min, region5max, region6min, region6max: real;
  inputerror: boolean;
  inputused: array[1..6] of boolean;
implementation
{$R *.DFM}
Procedure wavenumbers_size_maximumset; {Determine the number of data points that the
user is interested in}
var
   x: integer;                                    wavenumber: real;
begin
    assignfile (textfile,'wavenumber.dat');
    reset(textfile);
    x:= 0;
    readln(textfile,wavenumber);
    x:= x + 1;
    while ((wavenumber < maximumwavelength) and not(eof(textfile))) do
        begin
            readln(textfile,wavenumber);
            x:= x + 1;
        end;
    if wavenumber > maximumwavelength
      then datasize:= x-2
```

```
          else datasize:= x-1;
    closefile(textfile);
end;
Procedure wavenumbers_size_fileset; {Determine the number of data points that the file
supplies}
var
  x: integer;                                        wavenumber: real;
begin
    assignfile (textfile,'wavenumber.dat');
    reset(textfile);
    x:= 0;
    readln(textfile,wavenumber);
    x:= x + 1;
    while not(eof(textfile)) do
        begin
            readln(textfile,wavenumber);
            x:= x + 1;
        end;
    datasize:= x;
    closefile(textfile);
end;
Procedure initialise_arrays; {Specify the size of the arrays}
begin
    setlength(intensityarray, totalint, datasize);
    setlength(wavearray, datasize);
    setlength(minarray, totalint);
    setlength(areaarray, totalint, areaarraydimension);
    setlength(maxarray, totalint);
    setlength(namearray, totalint);
end;
Procedure assign_wavenumbers; {Assign values to the wavenumber array}
var
  x: integer;
begin
    assignfile (textfile,'wavenumber.dat');
    reset(textfile);
    for x:= 0 to (datasize-1) do
       begin
           readln(textfile,wavearray[x]);
       end;
    closefile(textfile);
end;
Procedure intensities; {Input the absorbance values}
var
  x, y, position: integer;                           tempstring: string;
begin
    assignfile (textfile,'outintensity.prn');
    reset(textfile);
    readln(textfile, tempstring);
    for x:= 0 to (totalint-2) do
       begin
           position:= pos(' ',tempstring);
           if position = 1
             then position:= pos(' ',copy(tempstring,2,1000000));
           namearray[x]:= copy(tempstring, 1, position);
           delete(tempstring,1,position);
       end;
    namearray[totalint-1]:= tempstring;
    for x:= 0 to (datasize-1) do
       begin
```

```
            for y:= 0 to (totalint - 1) do
                begin
                    read(textfile,intensityarray[y,x]);
                end;
        end;
    closefile(textfile);
end;
Procedure baseline; {Input Baseline parameters}
var
  x, y: integer;                                        tempstring: string;
begin
    assignfile (textfile,'baseline.dat');
    reset(textfile);
    readln(textfile, tempstring);
    for x:= 0 to 3 do
        begin
            for y:= 0 to (totalint - 1) do
                begin
                    read(textfile,basearray[y,x]);
                end;
        end;
    closefile(textfile);
end;
Procedure find_min; {Find minimum absorbance}
var x, y: integer;
begin
    for x:= 0 to (totalint -1) do
        begin
            minarray[x]:= intensityarray[x,0];
            for y:= 1 to (datasize-1) do
                begin
                    minarray[x]:= min(minarray[x],intensityarray[x,y]);
                end;
        end;
end;
Procedure find_max; {Find maximum absorbance}
var x, y: integer;
begin
    for x:= 0 to (totalint -1) do
        begin
            maxarray[x]:= intensityarray[x,0];
            for y:= 1 to (datasize-1) do
                begin
                    maxarray[x]:= max(maxarray[x],intensityarray[x,y]);
                end;
        end;
end;
Procedure normalise_intensity; {Change intensities so that all absorbance values are between
0 and 1}
var x, y: integer;
begin
    for x:= 0 to (totalint - 1) do
        begin
            for y:= 0 to (datasize-1) do
                begin
                    intensityarray[x,y]:= (intensityarray[x,y]-minarray[x])/(maxarray[x]-
minarray[x])*maxintensity;
                end;
        end;
end;
```

```
procedure calculate_total_area; {Calculate the current total area}
var x, y: integer;
begin
    for x:= 0 to (totalint -1) do
        begin
            areaarray[x,0]:= 0;
            for y:= 0 to (datasize - 2) do
                begin
                    areaarray[x,0]:= areaarray[x,0] - 0.5*(wavearray[y]-
wavearray[y+1])*(intensityarray[x,y]+intensityarray[x,y+1]-2*minarray[x]);
                end;
        end;
end;
Procedure normalise_area; {Insure that the minimum absorbance is 0 and that the total area
equals the desired normarea}
var x, y: integer;
begin
    for x:= 0 to (totalint - 1) do
        begin
            for y:= 0 to (datasize-1) do
                begin
                    intensityarray[x,y]:= normarea*((intensityarray[x,y]-minarray[x])/(areaarray[x,0]));
                end;
        end;
end;
Procedure baseline_correction; {Subtract the baseline}
var
  x, y: integer;
begin
    for x:= 0 to (totalint-1) do
        begin
            for y:= 0 to (datasize-1) do
                begin
                    intensityarray[x,y]:= intensityarray[x,y] -
basearray[x,0]*(wavearray[y]*wavearray[y]*wavearray[y]) -
basearray[x,1]*(wavearray[y]*wavearray[y])- basearray[x,2]*wavearray[y]- basearray[x,3];
                end;
        end;
end;
Procedure determine_intervals; {The starting wavenumbers for the six regions are defined}
var x: integer;
begin
    flag:= 0;
    For x:= 0 to (datasize-1) do
        begin
  if (wavearray[x] >= region1min) and (flag = 0)
            then
                begin
                    interval[flag]:= x;
                    flag:= flag + 1;
                end;

if (wavearray[x] >= region1max) and (flag = 1)
            then
                begin
                    interval[flag]:= x;
                    flag:= flag + 1;
                end;
 if (wavearray[x] >= region2min) and (flag = 2)
            then
                begin
                    interval[flag]:= x;
                    flag:= flag + 1;
                end;
if (wavearray[x] >= region2max) and (flag = 3)
            then
                begin
                    interval[flag]:= x;
                    flag:= flag + 1;
                end;
if (wavearray[x] >= region3min) and (flag = 4)
            then
                begin
                    interval[flag]:= x;
                    flag:= flag + 1;
                end;
if (wavearray[x] >= region3max) and (flag = 5)
            then
```

```
                begin
                    interval[flag]:= x;
                    flag:= flag + 1;
                end;
if (wavearray[x] >= region4min) and (flag = 6)
            then
                begin
                    interval[flag]:= x;
                    flag:= flag + 1;
                end;
if (wavearray[x] >= region4max) and (flag = 7)
            then
                begin
                    interval[flag]:= x;
                    flag:= flag + 1;
                end;
 if (wavearray[x] >= region5min) and (flag = 8)
            then
                begin
                    interval[flag]:= x;
                    flag:= flag + 1;
                end;
 if (wavearray[x] >= region5max) and (flag = 9)
            then
                begin
                    interval[flag]:= x;
                    flag:= flag + 1;
                end;
 if (wavearray[x] >= region6min) and (flag = 10)
            then
                begin
                    interval[flag]:= x;
                    flag:= flag + 1;
                end;
if (wavearray[x] >= region6max) and (flag = 11)
            then
                begin
                    interval[flag]:= x;
                    flag:= flag + 1;
                end;
    end;

end;
Procedure write_output; {Export the intensity normalised spectra}
var x, y, z: integer;
begin
  Assignfile (textfile,'outintensity.prn');
  Assignfile (textfile2, 'outwavenumber.dat');
  rewrite(textfile);
  rewrite(textfile2);
  for z:= 0 to (totalint -1) do
     begin
         write(textfile, namearray[z]);
         write(textfile, ' ');
     end;
  writeln(textfile);
  for x:= 0 to (datasize-1) do
begin
    for y:= 0 to (totalint-1) do
       begin
           write(textfile, intensityarray[y,x]);
           write(textfile, ' ');
       end;
    writeln(textfile);
    writeln(textfile2, wavearray[x]);
end;
closefile(textfile);
closefile(textfile2);
Procedure Mean Frequency; {Calculate the mean frequency}
var
x, y, flag: integer;
area: real;
begin
    for x:= 0 to (totalint-1) do
       begin
           flag:= 0;
           y:= 0;
           area:= 0;
           while flag = 0 do
                begin
                    area:= area - 0.5*(wavearray[y]-
wavearray[y+1])*(intensityarray[x,y]+intensityarray[x,y+1]-2*minarray[x]);
                    if (area > (0.5*areaarray[x,0]))
                    then
                       begin
                           areaarray[x,8]:= wavearray[y];
                           flag:= flag +1;
                       end;
                    y:= y + 1;
           end;
       end;
```

```
end;
procedure relative_areas; {Calculate the relative area in each region}
var x, y, n: integer;
begin
    for x:= 0 to (totalint - 1) do
       begin
          n:= 0;
          while n <= 11 do
             begin
                y:= interval[n];
                while y <= interval[n+1] do
                   begin
                   areaarray[x,round((n/2)+1)]:= areaarray[x,round((n/2)+1)]-0.5*(wavearray[y]-
wavearray[y+1])*(intensityarray[x,y]+intensityarray[x,y+1] - 2*minarray[x]);
                   y:= y + 1;
                   end;
                areaarray[x,round((n/2)+1)]:= areaarray[x,round((n/2)+1)]/areaarray[x,0]*100;
                n:= n + 2;
             end;
          areaarray[x,7]:= areaarray[x,topratioarea]/areaarray[x,bottomratioarea]; {ratio}
       end;
end;
Procedure write_relative_area_output; {Export the Area, ratio and mean frequency data}
var x, y: integer;
begin
  Assignfile (textfile,'areaoutput.dat');
  rewrite(textfile);
  writeln(textfile, 'Name Total Region1 Region2 Region3 Region4 Region5 Region6 Ratio Mean
Frequency');
  for x:= 0 to (totalint - 1) do                                    write(textfile, areaarray[x,y]);
     begin                                                          write(textfile, ' ');
        write(textfile, namearray[x]);                            end;
        write(textfile, ' ');                                   writeln(textfile,'')
        for y:= 0 to 8 do                                   end;
           begin                                            closefile(textfile);
end;
procedure TForm1.CalculateButtonClick(Sender: TObject);
begin
    inputerror:= false;
{Obtain the dimensions of the arrays based on wavenumber}
    if MaximumvalueCheckBox.Checked
      then
         begin
            val (maxwaveinput.text, maximumwavelength,err);
            wavenumbers_size_maximumset;
         end;
    if standard_area_calc_radio.Checked
      then
         begin
            maximumwavelength:= 2500;
            wavenumbers_size_maximumset;
            {define interval regions}
         end;
    if not(MaximumvalueCheckBox.Checked or standard_area_calc_radio.Checked)
      then wavenumbers_size_fileset;
{Obtain the dimensions of the arrays based on number of spectra}
    Val (numberofspectrainput.text,total,err);
    totalint:= round(total);
{Obtain the dimensions of the arrays based on whether the areaarray is used to normalise the
area or do a region analysis}
```

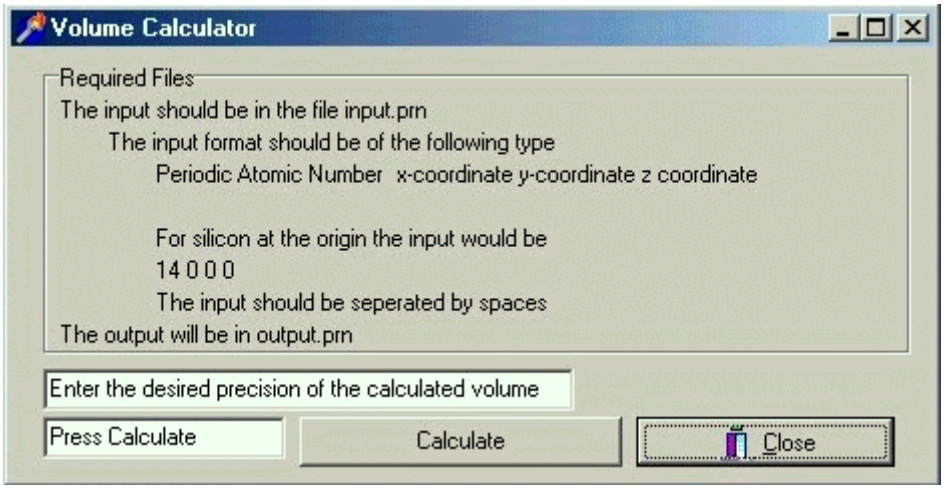
```
    if (costum_area_calc_radio.Checked or standard_area_calc_radio.Checked)
      then areaarraydimension:= 9
          else areaarraydimension:= 1;
{Input intensities into initialised arrays}
    initialise_arrays;                                    get_rid_of_extra_spaces_intensity;
    assign_wavenumbers;                                   intensities;
    fix_minus_intensity;
{Normalise intensity to 1}
    if Norm_int_radio.Checked                                  find_max;
      then                                                     normalise_intensity;
        begin                                                  write_output;
            maxintensity:= 1;                                  endbox.text:= 'Done';
            find_min;                                     end;
{Normalise intensity to another value}
    if othermaxint_radio.Checked                               find_max;
      then                                                     normalise_intensity;
        begin                                                  write_output;
            val                                                endbox.text:= 'Done';
(input_maxint.text,maxintensity,err);                     end;
            find_min;
{Area normalisation}
    if normarea_radio.Checked                                  find_max;
      then                                                     calculate_total_area;
        begin                                                  normalise_area;
            val                                                write_output;
(Norm_area_input.text,normarea,err);                           endbox.text:= 'Done';
            find_min;                                     end;
{Baseline correction and area normalisation}
    if base_area_corr_radio.Checked                            val
      then                                            (Norm_area_input2.text,normarea,err);
        begin                                                  find_min;
            fix_minus_base;                                    find_max;
            get_rid_of_extra_spaces_base;                      calculate_total_area;
            setlength (basearray, totalint, 4);                normalise_area;
            baseline;                                          write_output;
            baseline_correction;                               endbox.text:= 'Done';
                                                          end;
{Baseline correction without area normalisation}
    if Base_corr_radio.Checked                                 baseline;
      then                                                     baseline_correction;
        begin                                                  write_output;
            fix_minus_base;                                    endbox.text:= 'Done';
            get_rid_of_extra_spaces_base;                 end;
            setlength (basearray, totalint, 4);
{Baseline correction and intensity normalisation}
    if base_int_corr_radio.Checked                             val
      then                                            (input_maxint2.text,maxintensity,err);
        begin                                                  find_min;
            fix_minus_base;                                    find_max;
            get_rid_of_extra_spaces_base;                      normalise_intensity;
            setlength (basearray, totalint, 4);                write_output;
            baseline;                                          endbox.text:= 'Done';
            baseline_correction;                          end;
{Calculate predefined area ratios}
    if standard_area_calc_radio.Checked                        region2max:= 800;
      then                                                     region3min:= 801;
        begin                                                  region3max:= 1000;
            region1min:= 401;                                  region4min:= 1001;
            region1max:= 500;                                  region4max:= 1300;
            region2min:= 501;                                  region5min:= 1301;
```

```
            region5max:= 1700;
            region6min:= 1701;
            region6max:= 2498;
            toprationarea:= 2;
            bottomratioarea:= 1;
            determine_intervals;
            find_min;
{Calculate user defined area ratios}
    if costum_area_calc_radio.Checked
       then
          begin
    if region1_CheckBox.Checked
   then
   begin
    val (region1begin.text, region1min, err);
    val (region1end.text, region1max, err);
        if (region1min >= region1max)
       or (region1min < wavearray[0])
      or (region1max > wavearray[datasize-1])
         then inputerror:= true;
    inputused[1]:= true;
   end
   else
   begin
    region1min:= wavearray[1];
    region1max:= wavearray[0];
    inputused[1]:= false;
   end;
    if region2_CheckBox.Checked
   then
   begin
    val (region2begin.text, region2min, err);
    val (region2end.text, region2max, err);
        if (region2min >= region2max)
       or (region2min < wavearray[0])
      or (region2max > wavearray[datasize-1])
         then inputerror:= true;
    inputused[2]:= true;
   end
   else
   begin
    region2min:= wavearray[1];
    region2max:= wavearray[0];
    inputused[2]:= false;
   end;
    if region3_CheckBox.Checked
   then
   begin
    val (region3begin.text, region3min, err);
    val (region3end.text, region3max, err);
    if (region3min >= region3max)
       or (region3min < wavearray[0])
      or (region3max > wavearray[datasize-1])
         then inputerror:= true;
    inputused[3]:= true;
   end
   else
   begin
    region3min:= wavearray[1];
    region3max:= wavearray[0];
    inputused[3]:= false;
   end;
    if region4_CheckBox.Checked
   then
   begin
    val (region4begin.text, region4min, err);
    val (region4end.text, region4max, err);
    if (region4min >= region4max)
       or (region4min < wavearray[0])
      or (region4max > wavearray[datasize-1])
         then inputerror:= true;
    inputused[4]:= true;
   end
   else
   begin
    region4min:= wavearray[1];
    region4max:= wavearray[0];
    inputused[4]:= false;
   end;
    if region5_CheckBox.Checked
   then
   begin
    val (region5begin.text, region5min, err);
    val (region5end.text, region5max, err);
    if (region5min >= region5max)
       or (region5min < wavearray[0])
      or (region5max > wavearray[datasize-1])
         then inputerror:= true;
    inputused[5]:= true;
   end
   else
   begin
    region5min:= wavearray[1];
    region5max:= wavearray[0];
    inputused[5]:= false;
   end;
    if region6_CheckBox.Checked
   then
   begin
    val (region6begin.text, region6min, err);
    val (region6end.text, region6max, err);
    if (region6min >= region6max)
       or (region6min < wavearray[0])
      or (region6max > wavearray[datasize-1])
         then inputerror:= true;
    inputused[6]:= true;
   end
   else
   begin
            calculate_total_area;
            relative_areas;
            Mean Frequency;
            write_relative_area_output;
            endbox.text:= 'Done';
        end;
```

```
      region6min:= wavearray[1];
      region6max:= wavearray[0];
      inputused[6]:= false;
     end;
      val (numeratoregion.text, topratioareareal,
err);
      val (ratiodenominator.text,
bottomratioareareal, err);
      topratioarea:= round(topratioareareal);
      bottomratioarea:=
round(bottomratioareareal);
      if not(inputused[topratioarea] and
inputused[bottomratioarea])
    then inputerror:= true;
end;
end;
end.
```

```
      if not(inputerror)
     then
     begin
      determine_intervals;
      find_min;
      calculate_total_area;
      relative_areas;
      Mean Frequency;
      write_relative_area_output;
      endbox.text:= 'Done';
     end
      else endbox.text:= 'There is an error in the
input';
```

A2. VOLUME CALCULATION PROGRAM




```
{This program uses the Cartesian coordinates and atomic numbers of the input file
to calculate the volume based on the method of A Gavezzotti, J. Am. Chem. Soc. 105, 5220-
5225, 1983
The atomic radii are taken from Gavezzotti,
R.D. Shannon,   Acta Cryst. A32, 751-767,1976
A Bondi, J. Phys. Chem. 68(3), 441-451, 1964}
unit Vol_calc;
uses
  Windows, Messages, SysUtils, Classes, Graphics, Controls, Forms, Dialogs,
  StdCtrls, Buttons, Math;
type
  TForm1 = class(TForm)
    CloseBitBtn: TBitBtn;
    Calculate_button: TButton;
    files_GroupBox: TGroupBox;
    input_Label: TLabel;
    format_Label: TLabel;
    description_Label: TLabel;
    example_Label: TLabel;
    more_Label: TLabel;
    output_Label: TLabel;
    user_comm: TEdit;
    example_Label2: TLabel;
    inout2_Label: TLabel;
    Presicion: TEdit;
    procedure Calculate_buttonClick(Sender:
TObject);
  end;
var
```

```
  Form1: TForm1;
  coordinate_array: array of array of real;
  in_atom_array: array of array of array of real;
  VDWrad, Atom_rad: array of real;
  number_of_atoms, err, number_pointsx, number_pointsy, number_pointsz: integer;
  volume_total, volume_incr, startx, starty, startz: real;
  volume_atoms, volume_cavity, endx, endy, endz: real;
  flag_inside_atom, flag_in_cavity: boolean;
implementation
{$R *.DFM}
procedure input_vdwrad;
var
   textfile: text;                                          x: integer;
{Reads the van der Waals radii into the array}
begin;
      assignfile (textfile, 'vdw_rad.txt');                for x:= 0 to  102 do
      reset(textfile);                                           begin
      setlength(VDWrad, 103);                                       readln(textfile);
                                                                    readln(textfile,VDWrad[x]);
                                                                 end;
end;
procedure input_from_file;
var
   textfile: text;
   x, y: integer;
   atomnumber_real: real;
{Reads in the data from the input file: input.prn}
begin;
     assignfile (textfile,'input.prn');
     reset(textfile);
     {First determine the number of atoms}
     x:= 0;
     while not(eof(textfile)) do
          begin
              readln(textfile);
              x:= x + 1;
          end;
     number_of_atoms:= x-1;
     reset(textfile);
     {Initialise the dinamic arrays}
     setlength(coordinate_array, (number_of_atoms+1), 3);
     setlength(Atom_rad, (number_of_atoms+1));
     {Read the values into the appropriate arrays}
     for x:= 0 to number_of_atoms do
        begin
            read(textfile,atomnumber_real);
            Atom_rad[x]:= VDWrad[round(atomnumber_real)-1];
            for y:= 0 to 2 do
               begin
                   read(textfile,coordinate_array[x,y]);
               end;
            readln(textfile);
        end;
     closefile(textfile);
end;
procedure coordinate_system;
var
   x: integer;                                              maxrad: real;
{Determines the dimensions of the volume to be searched}
begin;
```

```
    startx:= 0;                                          endy:= 0;
    starty:= 0;                                          endz:= 0;
    startz:= 0;                                          maxrad:= 0;
    endx:= 0;
    for x:= 0 to number_of_atoms do
       begin
          endx:= max(endx,coordinate_array[x,0]);
          endy:= max(endy,coordinate_array[x,1]);
          endz:= max(endz,coordinate_array[x,2]);
          startx:= min(startx,coordinate_array[x,0]);
          starty:= min(starty,coordinate_array[x,1]);
          startz:= min(startz,coordinate_array[x,2]);
          maxrad:= max(maxrad,atom_rad[x]);
       end;
    {ensure that the molecule is in the correct octant of 3D space}
    for x:= 0 to number_of_atoms do
       begin
          coordinate_array[x,0]:= coordinate_array[x,0] + abs(startx) + 1.1*maxrad;
          coordinate_array[x,1]:= coordinate_array[x,1] + abs(starty) + 1.1*maxrad;
          coordinate_array[x,2]:= coordinate_array[x,2] + abs(startz) + 1.1*maxrad;
       end;
    number_pointsx:= trunc((2*1.1*maxrad + endx - startx)/volume_incr) + 1;
    number_pointsy:= trunc((2*1.1*maxrad + endy - starty)/volume_incr) + 1;
    number_pointsz:= trunc((2*1.1*maxrad + endz - startz)/volume_incr) + 1;
    setlength(in_atom_array, number_pointsx+1, number_pointsy+1, number_pointsz+1);
end;
Procedure initialise_in_atom_array;
var
  x, y, z:integer;
begin
   for x:= 0 to number_pointsx do
      begin
         for y:= 0 to number_pointsy do
            begin
               for z:= 0 to number_pointsz do
                  begin
                     in_atom_array[x,y,z]:= 0;
                  end;
            end;
      end;
end;
procedure calculate_volume;
var
  a, x, y, z, molendz:integer;
  posx, posy, posz, centerx, centery, centerz, sum_total, sum_cavity, sum_atoms : real;
  flag_inside_mol, flag_inside_wall: boolean;
begin;
    initialise_in_atom_array;
    for x:= 0 to number_pointsx do
      begin
         posx:= x*volume_incr;
         for y:= 0 to number_pointsy do
           begin
              posy:= y*volume_incr;
              for z:= 0 to number_pointsz do
                 begin
                    posz:= z*volume_incr;
                    flag_inside_atom:= false;
                    a:= 0;
                    while ((a <= number_of_atoms) and not(flag_inside_atom)) do
```

```
                        begin
                            centerx:= posx - coordinate_array[a,0];
                            centery:= posy - coordinate_array[a,1];
                            centerz:= posz - coordinate_array[a,2];
                            flag_inside_atom:= (((centerx*centerx)+(centery*centery)+(centerz*centerz))
<= (Atom_rad[a]*Atom_rad[a]));
                            if flag_inside_atom
                                then in_atom_array[x,y,z]:= 1;
                            a:= a +1;
                          end;
                    end;
              end;
        end;
    sum_total:= 0;
    sum_atoms:= 0;
    sum_cavity:= 0;
    for x:= 0 to number_pointsx do
       begin
          for y:= 0 to number_pointsy do
             begin
                flag_inside_mol:= false;
                flag_inside_wall:= false;
                molendz:= number_pointsz;
                z:= number_pointsz;
                while z >= 0 do
                   begin
                      if (in_atom_array[x,y,z] = 1)
                        then
                           begin
                              molendz:= z;
                              z:= 0;
                           end;
                      z:= z -1;
                   end;
                for z:= 1 to molendz do
                   begin
                      sum_atoms:= sum_atoms + in_atom_array[x,y,z];
                      if (in_atom_array[x,y,z] = 1)
                        then
                           begin
                              flag_inside_mol:= true;
                              flag_inside_wall:= true;
                           end;
                      if flag_inside_mol and not (in_atom_array[x,y,z] = in_atom_array[x,y,(z-1)])
                        then flag_inside_wall:= false;
                      if flag_inside_mol and not(flag_inside_wall)
                        then sum_cavity:= sum_cavity + 1;
                      if flag_inside_mol
                        then sum_total:= sum_total + 1;
                   end;
             end;
       end;
    volume_incr:= Power(volume_incr,3);                    volume_cavity:= sum_cavity*volume_incr;
    volume_total:= sum_total*volume_incr;                  volume_atoms:= sum_atoms*volume_incr;
end;
procedure output;
var
  x, y, z:integer;
  textfile: text;
{The results are saved in the file output.prn}
```

```
begin
    assignfile (textfile,'output.prn');
    rewrite(textfile);
    write(textfile, 'Total volume: ');
    writeln(textfile,volume_total);
    write(textfile, 'Volume ocuppied by the
atoms: ');
    writeln(textfile,volume_atoms);
    write(textfile, 'Cavity volume: ');
    writeln(textfile,volume_cavity);
    closefile(textfile);
end;
procedure TForm1.Calculate_buttonClick(Sender: TObject);
begin;
     input_vdwrad;
     input_from_file;
     Val(Presicion.text,volume_incr,err);
     coordinate_system;
     calculate_volume;
     output;
     user_comm.text:= 'Done';
end;
end.
```

B.RAW DATA

B1. ULTRAMARINE BLUE KINETIC PEAK FITTING DATA

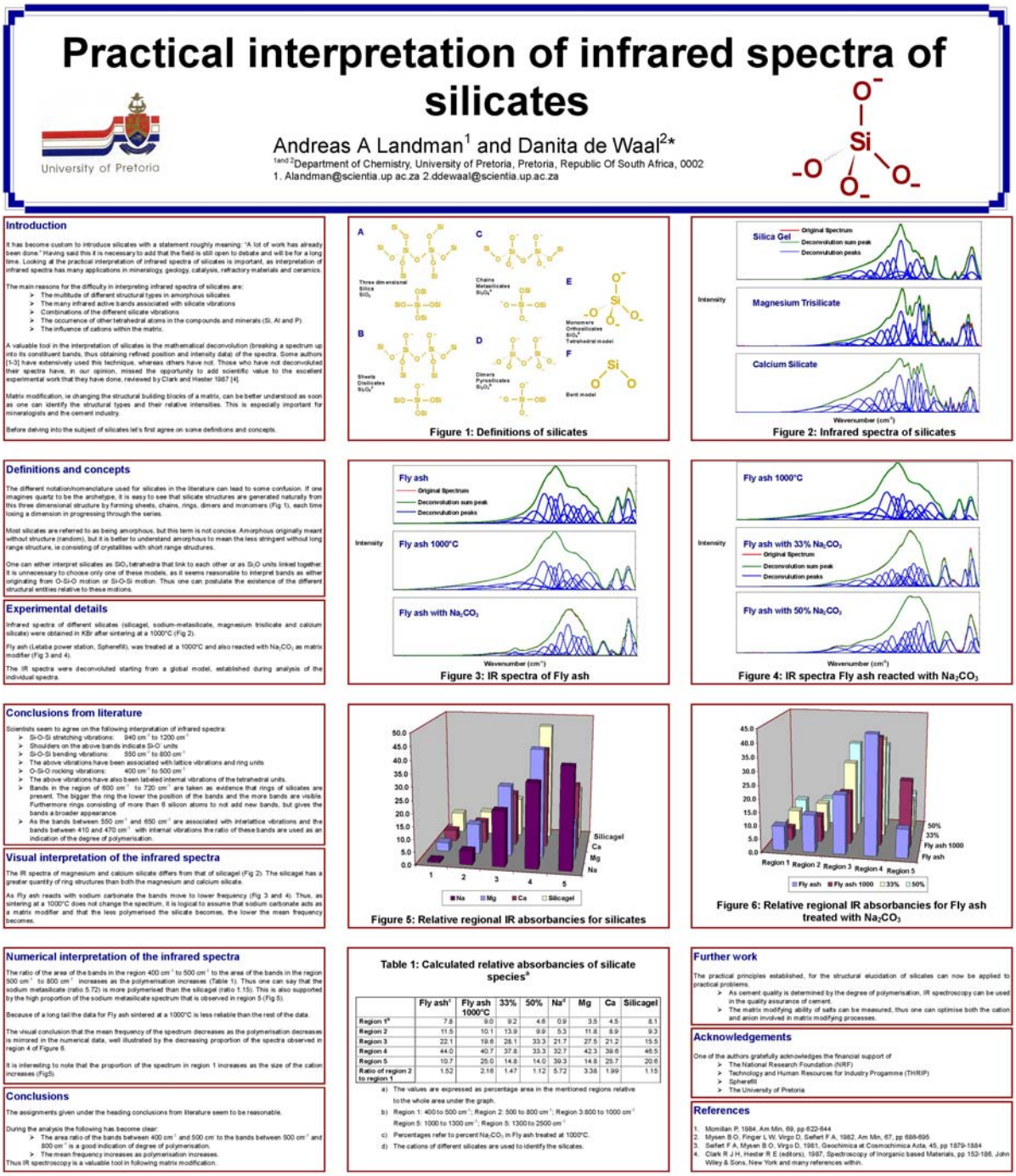
Spectrum	Peak 1	Peak 1 Int	Peak 2	Peak 2 Int	Ratio	Ave Ratio	Uncertainty
umbl10	542.725	3456.36	578.986	670.653	0.2		
umbl102	540.069	2283.56	574.995	958.82	0.4		
umbl103	543.762	1183.07	578.865	215.091	0.2	0.19	0.02
umbl104	541.757	1197.34	576.442	463.313	0.4	0.3	0.1
umbl301	546.424	193.663	585.666	21.9575	0.1		
umbl302	545.009	705.423	582.165	81.9958	0.1		
umbl303	544.169	1854.61	580.265	287.992	0.2		
umbl304	542.403	4272.75	578.351	857.254	0.2		
umbl305	543.625	1178.56	579.748	222.864	0.2	0.15	0.04
umbl60	544.527	2808.5	582.919	347.586	0.1		
umbl602	545.156	2785.09	583.421	314.326	0.1		
umbl603	544.237	4318.1	581.031	578.612	0.1		
umbl604	544.215	6725.48	581.998	851.471	0.1	0.12	0.01
umbl95	545.967	759.234	578.029	69.1982	0.1		
umbl953	546.469	1253.29	583.098	93.691	0.1		
umbl954	545.479	1483.57	582.583	157.149	0.1		
umbl955	544.44	4829.83	578.315	617.86	0.1	0.10	0.02
Umblack	543.982	1506.58	579.563	263.521	0.2		
Umblack2	543.945	3788.4	577.968	900.849	0.2		
Umblack3	543.943	11462.2	578.36	2456.75	0.2		
Umblack4	542.649	3574.16	579.635	853.635	0.2	0.22	0.03

C.RAW DATA FOR THE CALCULATION OF THE TRUNCATED OCTAHEDRON VOLUME IN SODALITE CRYSTALS

Authors	Large Base Å	Small Base Å	Diagonal Height Å	Height Å	Volume Å ^{3}
Werner1	6.273	4.435	4.959	3.841	217.1
Werner2	6.202	4.386	4.903	3.797	209.8
Werner3	6.156	4.353	4.867	3.770	205.2
Werner4	6.092	4.308	4.816	3.730	198.9
Flesche	6.234	4.408	4.928	3.817	213.1
Nilesen1	6.276	4.438	4.962	3.844	217.5
Nilesen2	6.316	4.466	4.993	3.867	221.6
Nilesen3	6.370	4.505	5.036	3.900	227.4
Mead1	6.362	4.498	5.029	3.895	226.5
Mead2	6.380	4.511	5.043	3.906	228.4
Mead3	6.363	4.500	5.030	3.896	226.6
Mead4	6.394	4.522	5.056	3.916	230.0
Lons	6.272	4.435	4.958	3.840	217.0

D.POSTERS

Landman AA, De Waal D, *Interpreting Infrared Spectra of Silicates*, proc SACI2000, 26 September 2000, p 138.



Practical interpretation of infrared spectra of silicates

University of Pretoria

Andreas A Landman[1] and Danita de Waal[2]*
[1 and 2]Department of Chemistry, University of Pretoria, Pretoria, Republic Of South Africa, 0002
1. Alandman@scientia.up.ac.za 2.ddewaal@scientia.up.ac.za

Introduction

It has become custom to introduce silicates with a statement roughly meaning: "A lot of work has already been done." Having said this it is necessary to add that the field is still open to debate and will be for a long time. Looking at the practical interpretation of infrared spectra of silicates is important, as interpretation of infrared spectra has many applications in mineralogy, geology, catalysis, refractory materials and ceramics.

The main reasons for the difficulty in interpreting infrared spectra of silicates are:

- The multitude of different structural types in amorphous silicates.
- The many infrared active bands associated with silicate vibrations
- Combinations of the different silicate vibrations
- The occurrence of other tetrahedral atoms in the compounds and minerals (Si, Al and P)
- The influence of cations within the matrix.

A valuable tool in the interpretation of silicates is the mathematical deconvolution (breaking a spectrum up into its constituent bands, thus obtaining refined position and intensity data) of the spectra. Some authors [1-3] have extensively used this technique, whereas others have not. Those who have not deconvoluted their spectra have, in our opinion, missed the opportunity to add scientific value to the excellent experimental work that they have done, reviewed by Clark and Hester 1987 [4].

Matrix modification, ie changing the structural building blocks of a matrix, can be better understood as soon as one can identify the structural types and their relative intensities. This is especially important for mineralogists and the cement industry.

Before delving into the subject of silicates let's first agree on some definitions and concepts.

Definitions and concepts

The different notation/nomenclature used for silicates in the literature can lead to some confusion. If one imagines quartz to be the archetype, it is easy to see that silicate structures are generated naturally from this three dimensional structure by forming sheets, chains, rings, dimers and monomers (Fig 1), each time losing a dimension in progressing through the series.

Most silicates are referred to as being amorphous, but this term is not concise. Amorphous originally meant without structure (random), but it is better to understand amorphous to mean the less stringent without long range structure, ie consisting of crystallites with short range structures.

One can either interpret silicates as SiO_4 tetrahedra that link to each other or as Si-O units linked together. It is unnecessary to choose only one of these models, as it seems reasonable to interpret bands as either originating from O-Si-O motion or Si-O-Si motion. Thus one can postulate the existence of the different structural entities relative to these motions.

Experimental details

Infrared spectra of different silicates (silicagel, sodium-metasilicate, magnesium trisilicate and calcium silicate) were obtained in KBr after sintering at a 1000°C (Fig 2).

Fly ash (Letaba power station, Spherefill), was treated at a 1000°C and also reacted with Na_2CO_3 as matrix modifier (Fig 3 and 4).

The IR spectra were deconvoluted starting from a global model, established during analysis of the individual spectra.

Conclusions from literature

Scientists seem to agree on the following interpretation of infrared spectra:

- Si-O-Si stretching vibrations: 940 cm^{-1} to 1200 cm^{-1}
- Shoulders on the above bands indicate Si-O$^-$ units
- Si-O-Si bending vibrations: 550 cm^{-1} to 800 cm^{-1}
- The above vibrations have been associated with lattice vibrations and ring units
- O-Si-O rocking vibrations: 400 cm^{-1} to 500 cm^{-1}
- The above vibrations have also been labeled internal vibrations of the tetrahedral units.
- Bands in the region of 600 cm^{-1} to 720 cm^{-1} are taken as evidence that rings of silicates are present. The bigger the ring the lower the position of the bands and the more bands are visible. Furthermore rings consisting of more than 6 silicon atoms to not add new bands, but gives the bands a broader appearance.
- As the bands between 550 cm^{-1} and 650 cm^{-1} are associated with interlattice vibrations and the bands between 410 and 470 cm^{-1} with internal vibrations the ratio of these bands are used as an indication of the degree of polymerisation.

Visual interpretation of the infrared spectra

The IR spectra of magnesium and calcium silicate differs from that of silicagel (Fig 2). The silicagel has a greater quantity of ring structures than both the magnesium and calcium silicate.

As Fly ash reacts with sodium carbonate the bands move to lower frequency (Fig 3 and 4). Thus, as sintering at a 1000°C does not change the spectrum, it is logical to assume that sodium carbonate acts as a matrix modifier and that the less polymerised the silicate becomes, the lower the mean frequency becomes.

Numerical interpretation of the infrared spectra

The ratio of the area of the bands in the region 400 cm^{-1} to 500 cm^{-1} to the area of the bands in the region 900 cm^{-1} to 800 cm^{-1} increases as the polymerisation increases (Table 1). Thus one can say that the sodium metasilicate (ratio 5.72) is more polymerised than the silicagel (ratio 1.15). This is also supported by the high proportion of the sodium metasilicate spectrum that is observed in region 5 (Fig 5).

Because of a long tail the data for Fly ash sintered at a 1000°C is less reliable than the rest of the data.

The visual conclusion that the mean frequency of the spectrum decreases as the polymerisation decreases is mirrored in the numerical data, well illustrated by the decreasing proportion of the spectra observed in region 4 of Figure 6.

It is interesting to note that the proportion of the spectrum in region 1 increases as the size of the cation increases (Fig 5).

Conclusions

The assignments given under the heading conclusions from literature seem to be reasonable.

During the analysis the following has become clear:

- The area ratio of the bands between 400 cm^{-1} and 500 cm^{-1} to the bands between 500 cm^{-1} and 800 cm^{-1} is a good indication of degree of polymerisation.
- The mean frequency increases as polymerisation increases.

Thus IR spectroscopy is a valuable tool in following matrix modification.

Figure 1: Definitions of silicates

Figure 2: Infrared spectra of silicates

Figure 3: IR spectra of Fly ash

Figure 4: IR spectra Fly ash reacted with Na_2CO_3

Figure 5: Relative regional IR absorbancies for silicates

Figure 6: Relative regional IR absorbancies for Fly ash treated with Na_2CO_3

Table 1: Calculated relative absorbancies of silicate species[a]

	Fly ash[c]	Fly ash 1000°C	33%	50%	Na[d]	Mg	Ca	Silicagel
Region 1[b]	7.8	9.0	9.2	4.8	0.9	3.5	4.5	8.1
Region 2	11.5	10.1	13.9	9.9	5.3	11.8	8.9	9.3
Region 3	22.1	19.8	28.1	33.3	21.7	27.5	21.2	15.5
Region 4	44.0	40.7	37.8	33.3	32.7	42.3	39.6	45.5
Region 5	10.7	25.0	14.8	14.0	39.3	14.8	25.7	20.6
Ratio of region 2 to region 1	1.52	2.18	1.47	1.12	5.72	3.36	1.99	1.15

a) The values are expressed as percentage area in the mentioned regions relative to the whole area under the graph.
b) Region 1: 400 to 500 cm^{-1}; Region 2: 500 to 800 cm^{-1}; Region 3 800 to 1000 cm^{-1} Region 5: 1000 to 1300 cm^{-1}; Region 5: 1300 to 2500 cm^{-1}
c) Percentages refer to percent Na_2CO_3 in Fly ash treated at 1000°C.
d) The cations of different silicates are used to identify the silicates.

Further work

The practical principles established, for the structural elucidation of silicates can now be applied to practical problems.

- As cement quality is determined by the degree of polymerisation, IR spectroscopy can be used in the quality assurance of cement.
- The matrix modifying ability of salts can be measured, thus one can optimise both the cation and anion involved in matrix modifying processes.

Acknowledgements

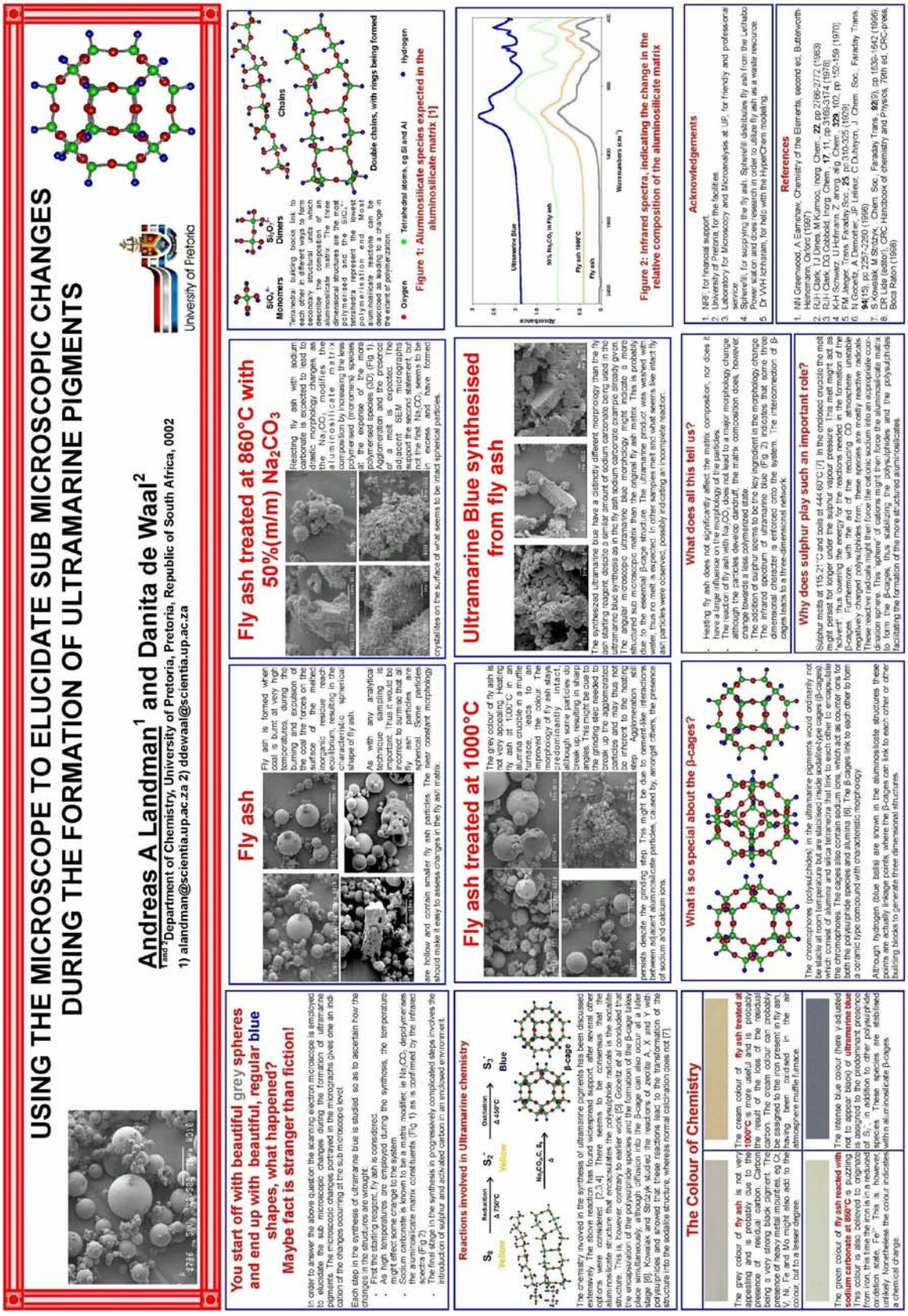
One of the authors gratefully acknowledges the financial support of

- The National Research Foundation (NRF)
- Technology and Human Resources for Industry Progamme (THRIP)
- Spherefill
- The University of Pretoria

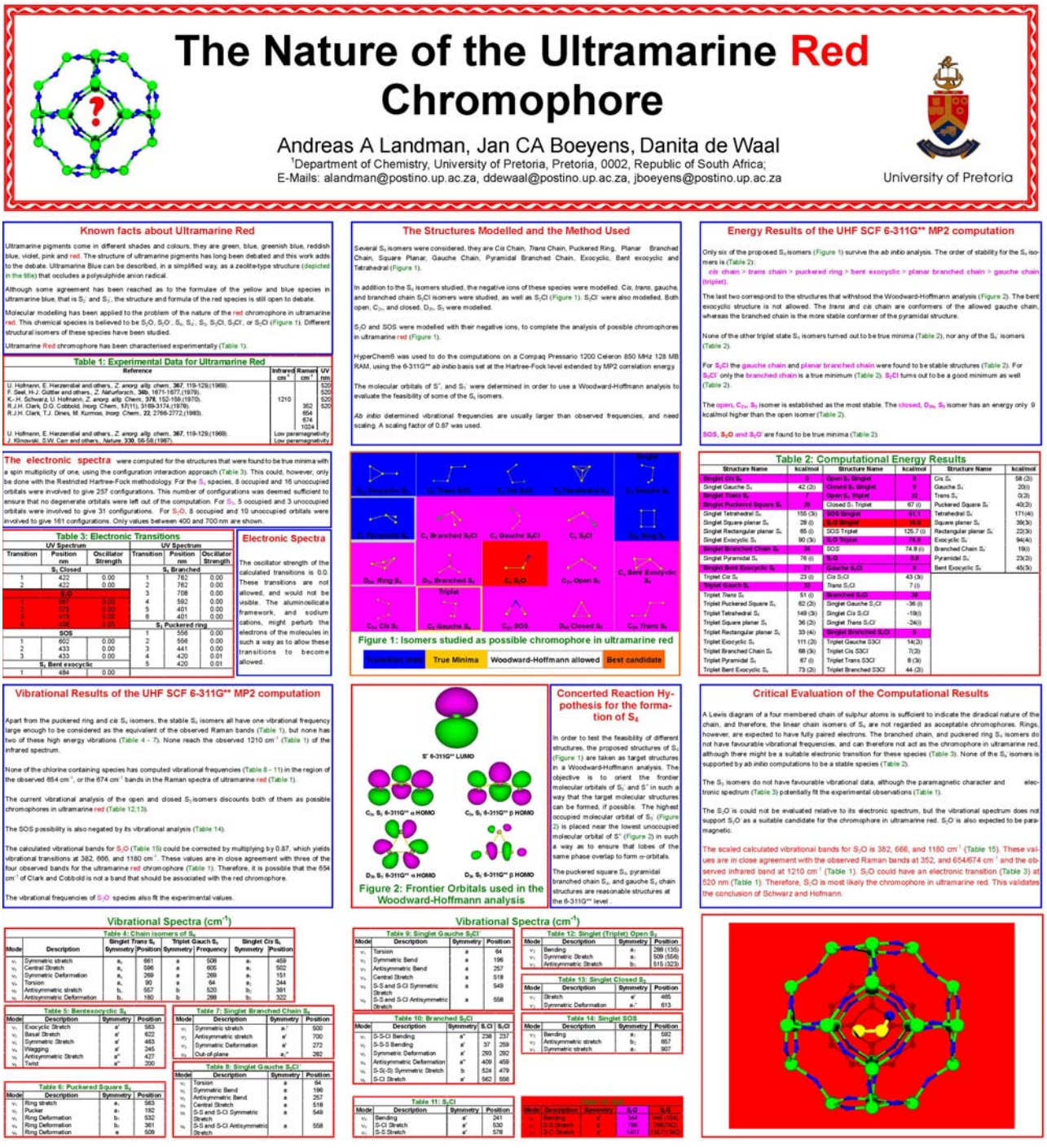
References

1. Momillan P, 1984, Am Min, 69, pp 622-644
2. Mysen B O, Finger L W, Virgo D, Seifert F A, 1982, Am Min, 67, pp 686-695
3. Seifert F A, Mysen B O, Virgo D, 1981, Geochimica et Cosmochimica Acta, 45, pp 1879-1884
4. Clark R J H, Hester R E (editors), 1987, Spectroscopy of Inorganic based Materials, pp 153-186, John Wiley & Sons, New York and many references within.

Landman AA, De Waal, *Using The Microscope To Elucidate Sub Microscopic Changes During The Formation Of Ultramarine Pigments*, December 2001, proc Microscopic Society of Southern Africa, 31, p 37, 2001.



Landman AA, Boeyens JCA, De Waal D, *The Nature of the Ultramarine Red Chromophore*, 7th Annual Materials Modelling Meeting, University of the North, Polokwane, 2003.



The Nature of the Ultramarine Red Chromophore

Andreas A Landman, Jan CA Boeyens, Danita de Waal
[1]Department of Chemistry, University of Pretoria, Pretoria, 0002, Republic of South Africa;
E-Mails: alandman@postino.up.ac.za, ddewaal@postino.up.ac.za, jboeyens@postino.up.ac.za
University of Pretoria

Known facts about Ultramarine Red

Ultramarine pigments come in different shades and colours, they are green, blue, greenish blue, reddish blue, violet, pink and red. The structure of ultramarine pigments has long been debated and this work adds to the debate. Ultramarine Blue can be described, in a simplified way, as a zeolite-type structure (depicted in the title) that occludes a polysulphide anion radical.

Although some agreement has been reached as to the formulae of the yellow and blue species in ultramarine blue, that is S_2^- and S_3^-, the structure and formula of the red species is still open to debate.

Molecular modelling has been applied to the problem of the nature of the red chromophore in ultramarine red. This chemical species is believed to be S_2O, S_2O^-, S_n, S_4^-, S_2, S_2Cl, S_3Cl, or S_2Cl (Figure 1). Different structural isomers of these species have been studied.

Ultramarine Red chromophore has been characterised experimentally (Table 1).

Table 1: Experimental Data for Ultramarine Red

Reference	Infrared cm^{-1}	Raman cm^{-1}	UV nm
U. Hofmann, E. Herzenstiel and others, *Z. anorg. allg. chem.*, **267**, 119-129;(1969). F. Seel, H-J. Güttler and others, *Z. Naturforsch.*, **34b**, 1671-1677,(1979). K.-H. Schwarz,U. Hofmann, *Z. anorg. allg. Chem.*, **378**, 152-159,(1970). R.J.H. Clark, D.G. Cobbold, *Inorg. Chem.*, **17**(11), 3169-3174,(1978). R.J.H. Clark, T.J. Dines, M. Kurmoo, *Inorg. Chem.*, **22**, 2766-2772,(1983).	1210	352 654 674 1024	520 520 520
U. Hofmann, E. Herzenstiel and others, *Z. anorg. allg. chem.*, **267**, 119-129;(1968). J. Klinowski, S.W. Carr and others, *Nature*, **330**, 56-58;(1987).		Low paramagnetivity	Low paramagnetivity

The Structures Modelled and the Method Used

Several S_4 isomers were considered, they are Cis Chain, Trans Chain, Puckered Ring, Planar Branched Chain, Square Planar, Gauche Chain, Pyramidal Branched Chain, Exocyclic, Bent exocyclic and Tetrahedral (Figure 1).

In addition to the S_4 isomers studied, the negative ions of these species were modelled. *Cis*, *trans*, gauche, and branched chain S_3Cl isomers were studied, as well as S_2Cl (Figure 1). S_2Cl^- were also modelled. Both open, C_{2v}, and closed, D_{2h}, S_5 were modelled.

S_2O and SOS were modelled with their negative ions, to complete the analysis of possible chromophores in ultramarine red (Figure 1).

HyperChem6 was used to do the computations on a Compaq Pressario 1200 Celeron 850 MHz 128 MB RAM, using the 6-311G** *ab initio* basis set at the Hartree-Fock level extended by MP2 correlation energy.

The molecular orbitals of S^- and S_3^- were determined in order to use a Woodward-Hoffmann analysis to evaluate the feasibility of some of the S_4 isomers.

Ab initio determined vibrational frequencies are usually larger than observed frequencies, and need scaling. A scaling factor of 0.87 was used.

Energy Results of the UHF SCF 6-311G** MP2 computation

Only six of the proposed S_4 isomers (Figure 1) survive the *ab initio* analysis. The order of stability for the S_4 isomers is (Table 2):

cis chain > trans chain > puckered ring > bent exocyclic > planar branched chain > gauche chain (triplet).

The last two correspond to the structures that withstood the Woodward-Hoffmann analysis (Figure 2). The bent exocyclic structure is not allowed. The *trans* and *cis* chain are conformers of the allowed gauche chain, whereas the branched chain is the more stable conformer of the pyramidal structure.

None of the other triplet state S_4 isomers turned out to be true minima (Table 2), nor any of the S_4^- isomers (Table 2).

For S_2Cl the gauche chain and planar branched chain were found to be stable structures (Table 2). For S_3Cl only the branched chain is a true minimum (Table 2). S_2Cl turns out to be a good minimum as well (Table 2).

The open, C_{2v}, S_5 isomer is established as the most stable. The closed, D_{2h}, S_5 isomer has an energy only 9 kcal/mol higher than the open isomer (Table 2).

SOS, S_2O and S_2O^- are found to be true minima (Table 2).

The electronic spectra

were computed for the structures that were found to be true minima with a spin multiplicity of one, using the configuration interaction approach (Table 3). This could, however, only be done with the Restricted Hartree-Fock methodology. For the S_n species, 8 occupied and 16 unoccupied orbitals were involved to give 257 configurations. This number of configurations was deemed sufficient to ensure that no degenerate orbitals were left out of the computation. For S_5, 5 occupied and 3 unoccupied orbitals were involved to give 31 configurations. For S_2O, 8 occupied and 10 unoccupied orbitals were involved to give 161 configurations. Only values between 400 and 700 nm are shown.

Table 3: Electronic Transitions

UV Spectrum Transition	Position nm	Oscillator Strength	UV Spectrum Transition	Position nm	Oscillator Strength
S_2 Closed			**S_4 Branched**		
1	422	0.00	1	762	0.00
2	422	0.00	2	762	0.00
S_2O			3	708	0.00
1	667	0.00	4	592	0.00
2	579	0.00	5	401	0.00
3	415	0.00	6	401	0.00
4	408	0.00	**S_5 Puckered ring**		
SOS			1	556	0.00
1	602	0.00	2	556	0.00
2	433	0.00	3	441	0.00
3	433	0.00	4	420	0.01
S_4 Bent exocyclic			5	420	0.01
1	484	0.00			

Electronic Spectra

The oscillator strength of the calculated transitions is 0.0. These transitions are not allowed, and would not be visible. The aluminosilicate framework, and sodium cations, might perturb the electrons of the molecules in such a way as to allow these transitions to become allowed.

Figure 1: Isomers studied as possible chromophore in ultramarine red

Table 2: Computational Energy Results

Structure Name	kcal/mol	Structure Name	kcal/mol	Structure Name	kcal/mol
Singlet *Cis* S_4	0	Open S_5 Singlet	0	*Cis* S_4^-	58 (2)
Singlet Gauche S_4	42 (3)	Closed S_5 Singlet	9	Gauche S_4^-	20(i)
Singlet *Trans* S_4	7	Open S_5 Triplet	32	Trans S_4^-	0(2)
Singlet Puckered Square S_4	29	Closed S_5 Triplet	67 (i)	Puckered Square S_4^-	40(2)
Singlet Tetrahedral S_4	155 (3i)	SOS Singlet	62.1	Tetrahedral S_4^-	171(4i)
Singlet Square planar S_4	28 (i)	S_2O Singlet	15.4	Square planar S_4^-	39(3i)
Singlet Rectangular planar S_4	65 (i)	SOS Triplet	125.7 (i)	Rectangular planar S_4^-	22(3i)
Singlet Exocyclic S_4	90 (3i)	S_2O Triplet	74.9	Exocyclic S_4^-	94(4i)
Singlet Branched Chain S_4	29	SOS	74.9 (i)	Branched Chain S_4^-	19(i)
Singlet Pyramidal S_4	76 (i)	S_2O	9.0	Pyramidal S_4^-	23(2i)
Singlet Bent Exocyclic S_4	21	Gauche S_2Cl	0	Bent Exocyclic S_4^-	45(3i)
Triplet *Cis* S_4	23 (i)	*Cis* S_2Cl	43 (3i)		
Triplet Gauch S_4	32	*Trans* S_2Cl	7 (i)		
Triplet *Trans* S_4	51 (i)	Branched S_2Cl	0		
Triplet Puckered Square S_4	62 (2i)	Singlet Gauche S_3Cl	-36 (i)		
Triplet Tetrahedral S_4	149 (3i)	Singlet *Cis* S_3Cl	-19(i)		
Triplet Square planar S_4	36 (2i)	Singlet *Trans* S_3Cl	-24(i)		
Triplet Rectangular planar S_4	33 (4i)	Singlet Branched S_3Cl	0		
Triplet Exocyclic S_4	111 (2i)	Triplet Gauche $S3Cl$	14(2i)		
Triplet Branched Chain S_4	68 (3i)	Triplet *Cis* $S3Cl$	7(2i)		
Triplet Pyramidal S_4	67 (i)	Triplet Trans $S3Cl$	8 (3i)		
Triplet Bent Exocyclic S_4	73 (2i)	Triplet Branched $S3Cl$	44 (2i)		

Vibrational Results of the UHF SCF 6-311G** MP2 computation

Apart from the puckered ring and *cis* S_4 isomers, the stable S_4 isomers all have one vibrational frequency large enough to be considered as the equivalent of the observed Raman bands (Table 1), but none has two of these high energy vibrations (Table 4 - 7). None reach the observed 1210 cm^{-1} (Table 1) of the infrared spectrum.

None of the chlorine containing species has computed vibrational frequencies (Table 8 - 11) in the region of the observed 654 cm^{-1}, or the 674 cm^{-1} bands in the Raman spectra of ultramarine red (Table 1).

The current vibrational analysis of the open and closed S_5 isomers discounts both of them as possible chromophores in ultramarine red (Table 12,13).

The SOS possibility is also negated by its vibrational analysis (Table 14).

The calculated vibrational bands for S_2O (Table 15) could be corrected by multiplying by 0.87, which yields vibrational transitions at 382, 666, and 1180 cm^{-1}. These values are in close agreement with three of the four observed bands for the ultramarine red chromophore (Table 1). Therefore, it is possible that the 654 cm^{-1} of Clark and Cobbold is not a band that should be associated with the red chromophore.

The vibrational frequencies of S_2O species also fit the experimental values.

Figure 2: Frontier Orbitals used in the Woodward-Hoffmann analysis

Concerted Reaction Hypothesis for the formation of S_4

In order to test the feasibility of different structures, the proposed structures of S_4 (Figure 1) are taken as target structures in a Woodward-Hoffmann analysis. The objective is to orient the frontier molecular orbitals of S_3^- and S^+ in such a way that the target molecular structures can be formed, if possible. The highest occupied molecular orbital of S_3^- (Figure 2) is placed near the lowest unoccupied molecular orbital of S^+ (Figure 2) in such a way as to ensure that lobes of the same phase overlap to form σ-orbitals.

The puckered square S_4, pyramidal branched chain S_4, and gauche S_4 chain structures are reasonable structures at the 6-311G** level.

Critical Evaluation of the Computational Results

A Lewis diagram of a four membered chain of sulphur atoms is sufficient to indicate the diradical nature of the chain, and therefore, the linear chain isomers of S_4 are not regarded as acceptable chromophores. Rings, however, are expected to have fully paired electrons. The branched chain, and puckered ring S_4 isomers do not have favourable vibrational frequencies, and can therefore not act as the chromophore in ultramarine red, although there might be a suitable electronic transition for these species (Table 3). None of the S_4 isomers is supported by *ab initio* computations to be a stable species (Table 2).

The S_5 isomers do not have favourable vibrational data, although the paramagnetic character and electronic spectrum (Table 3) potentially fit the experimental observations (Table 1).

The S_2O is could not be evaluated relative to its electronic spectrum, but the vibrational spectrum does not support S_2O^- as a suitable candidate for the chromophore in ultramarine red. S_2O is also expected to be paramagnetic.

The scaled calculated vibrational bands for S_2O is 382, 666, and 1180 cm^{-1} (Table 15). These values are in close agreement with the observed Raman bands at 352, and 654/674 cm^{-1} and the observed infrared band at 1210 cm^{-1} (Table 1). S_2O could have an electronic transition (Table 3) at 520 nm (Table 1). Therefore, S_2O is most likely the chromophore in ultramarine red. This validates the conclusion of Schwarz and Hofmann.

Vibrational Spectra (cm^{-1})

Table 4: Chain isomers of S_4

Mode	Description	Singlet *Trans* S_4 Symmetry	Singlet *Trans* S_4 Position	Triplet Gauch S_4 Symmetry	Triplet Gauch S_4 Frequency	Singlet *Cis* S_4 Symmetry	Singlet *Cis* S_4 Position
ν_1	Symmetric stretch	a_g	661	a	508	a_1	458
ν_2	Central Stretch	a_g	596	a	605	a_1	502
ν_3	Symmetric Deformation	a_g	269	a	269	a_1	151
ν_4	Torsion	a_u	90	a	64	a_2	244
ν_5	Antisymmetric stretch	b_u	557	b	520	b_1	391
ν_6	Antisymmetric Deformation	b_u	180	b	288	b_1	322

Table 5: Benteexocyclic S_4

Mode	Description	Symmetry	Position
ν_1	Exocyclic Stretch	a'	563
ν_2	Basal Stretch	a'	622
ν_3	Symmetric Stretch	a'	463
ν_4	Wagging	a'	245
ν_5	Antisymmetric Stretch	a''	427
ν_6	Twist	a''	200

Table 6: Puckered Square S_4

Mode	Description	Symmetry	Position
ν_1	Ring stretch	a_1	563
ν_2	Pucker	a_2	192
ν_3	Ring Deformation	b_1	532
ν_4	Ring Deformation	b_2	361
ν_5	Ring Deformation	e	509

Table 7: Singlet Branched Chain S_4

Mode	Description	Symmetry	Position
ν_1	Symmetric stretch	a_1'	500
ν_2	Antisymmetric stretch	a''	700
ν_3	Symmetric Deformation	a'	272
ν_4	Out-of-plane	a_1''	282

Table 8: Singlet Gauche S_4Cl

Mode	Description	Symmetry	Position
ν_1	Torsion	a	64
ν_2	Symmetric Bend	a	196
ν_3	Antisymmetric Bend	a	257
ν_4	Central Stretch	a	518
ν_5	S-S and S-Cl Symmetric Stretch	a	549
ν_6	S-S and S-Cl Antisymmetric Stretch	a	558

Table 9: Singlet Gauche S_3Cl

Mode	Description	Symmetry	Position
ν_1	Torsion	a	64
ν_2	Symmetric Bend	a	196
ν_3	Antisymmetric Bend	a	257
ν_4	Central Stretch	a	518
ν_5	Symmetric Deformation	a	549
ν_6	S-S and S-Cl Symmetric Stretch	a	558

Table 10: Branched S_3Cl

Mode	Description	Symmetry	S_3Cl	S_3Cl
ν_1	S-S-Cl Bending	a''	238	237
ν_2	S-S-S Bending	a'	37	259
ν_3	Symmetric Deformation	a'	293	292
ν_4	Antisymmetric Deformation	a''	409	459
ν_5	S-S-Cl Symmetric Stretch	b	534	479
ν_6	S-Cl Stretch	a'	562	556

Table 11: S_2Cl

Mode	Description	Symmetry	Position
ν_1	Bending	a'	241
ν_2	S-Cl Stretch	a'	530
ν_3	S-S Stretch	a'	578

Table 12: Singlet (Triplet) Open S_5

Mode	Description	Symmetry	Position
ν_1	Bending	a_1	288 (135i)
ν_2	Symmetric Stretch	a_1	509 (564i)
ν_3	Antisymmetric Stretch	b_2	515 (323i)

Table 13: Singlet Closed S_5

Mode	Description	Symmetry	Position
ν_1	Stretch	a_1'	466
ν_2	Symmetric Deformation	a_1''	613

Table 14: Singlet SOS

Mode	Description	Symmetry	Position
ν_1	Bending	a_1	582
ν_2	Antisymmetric stretch	b_2	657
ν_3	Symmetric stretch	a_1	907

Table 15: S_2O

Mode	Description	Symmetry	S_2O	S_2O
ν_1	Bending	a'	344	382 (150i)
ν_2	S-S Stretch	a'	786	666 (74i)
ν_3	S-O Stretch	a'	1401	1180 (194i)

Landman AA, De Waal D, *Vanadium in the Mullite Structure*, SACI Inorganic '03, Roode Vallei Country Lodge, Pretoria, 2003.

Vanadium in the Mullite structure

Andreas A Landman, Danita de Waal

Department of Chemistry, University of Pretoria, Pretoria, 0002, alandman@postino.up.ac.za, ddewaal@postino.up.ac.za

What is Fly Ash?

Fly ash is obtained by separating the solids from the flue gases of furnaces fired with pulverised coal and is the predominantly inorganic residue of the minerals entrained in the coal.

Sphere-Fill (Pty) Ltd size-classifies the fly ash that it obtains from the Lethabo (a Tswana word, which means "good living" or "happiness" [1]) Power Station in the Northern Free State. The different size fractions are obtained by pneumatic control within cyclones at the power plant, a process described as air 'classification'.[2] The class F fly ash particles were of similar size: in the range of 1 μm to 12 μm, with an average diameter of less than 5 μm.

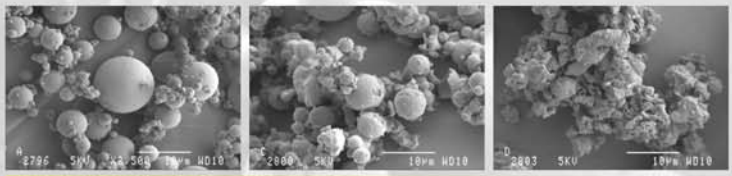


Figure 1: Scanning electron micrographs of fly ash, fly ash treated at 1 000 ºC, and fly ash reacted with ammonium metavanadate at 1 000 ºC

Synthesis Procedures

Fly ash was heated to 1000 °C in an alumina crucible for 24 hours to establish the effect of heating on the microscopic and sub-microscopic structure of fly ash.

Fly ash has a grey colour. After the heat treatment fly ash had a cream colour.

Ammonium metavanadate AR (Associated Chemical Enterprises cc) is sparsely soluble in water and was dissolved in a 5 % nitric acid CP (Chemical Suppliers (Pty) Ltd) solution. The vanadium species were forced to precipitate by evaporation unto the insoluble fly ash. The obtained powder was milled and heated at 1000 °C in an alumina crucible for 24 hours. Samples were annealed after a first round of analyses at 600 ºC for 24 hours, to obtain better Raman spectra.

The resulting products had, at some concentrations of vanadia (5, 7, 10 %), a yellow colour. This result was unexpected since the decomposition product of ammonium metavanadate is vanadium pentoxide, an orange-brown chemical species. The orange-brown colour was obtained at high concentrations of vanadia (25 %). In vanadium pentoxide the vanadium atom is found in a distorted octahedron (Title structure).

Ammonium metavanadate AR was placed in a porcelain dish with aluminium nitrate CP (uniLab, SAARCHEM (Pty) Ltd) and silicic acid (precipitated extra pure light, Merck). The mixture was heated at approximately 140 ºC in the presence of the water of hydration. At this stage the nitrate started to decompose. The dry fine solid was ground in a mortar and pestle and calcined in an alumina crucible at 600 ºC in a muffle furnace for 1 hour, to remove the remaining waters of hydration. After grinding the mixture again, the mixture was placed in the furnace and calcined at 1000 ºC for 24 hours.

The silicate-alumina-vanadia mixtures had a yellow colour, surpassing the yellow colour of fly ash reacted with ammonium metavanadate.

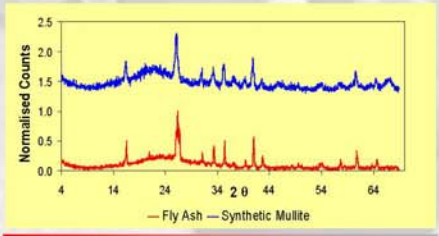


Figure 2: X-ray diffraction patterns

Scanning Electron Microscopy

Fly ash has a predominantly spherical microscopic structure (Fig 1). Upon heating fly ash, the scanning electron microscopy micrographs changed only slightly.

Scanning Electron Micrographs showed that there was more than just a physical interaction between the aluminosilicate and vanadia. The spherical shape of fly ash was destroyed during the reaction of fly ash with ammonium metavanadate. This reflected the fluxing nature of the vanadia species. The skeleton-like structures observed in the scanning electron microscopy micrographs were similar to the micrographs of HF etched fly ash particles.[3]

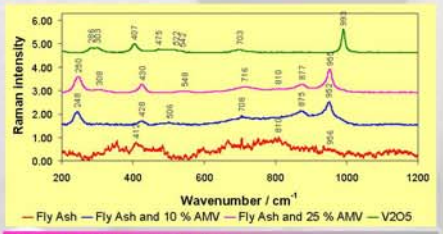


Figure 3: Raman spectra

X-Ray Diffraction

In the current X-ray diffraction patterns only mullite and quartz could be identified (Fig 2).[4,5] The quartz peak is not clearly visible in the presented pattern.

Neither the heat treatment nor reaction with ammonium metavanadate caused significant changes in the X-ray diffraction patterns of fly ash. **This might indicate that the vanadium atom replaced either a silicon or an aluminium atom in the aluminosilicate structure**, or that the vanadia species was not detectable.

Some synthesised aluminosilicates, of which only the synthesis of mullite is presented in this poster, gave a yellow colour. X-ray diffraction powder analyses identified the successful **aluminosilicate species as mullite** (Fig 2).

Raman Spectroscopy

Due to the combination of the small size and spherical microscopic structure of fly ash, no sensible Raman spectrum could be obtained for fly ash. The observed bands for fly ash treated with 10 % (a yellow product) and 25 % (a brown product) ammonium metavanadate (AMV) are similar to the bands for vanadium pentoxide (Fig 3). **This indicated that the vanadium could be in a distorted octahedral environment, as in vanadium pentoxide.**

Mullite Structure and Vanadium

Several types of interaction between the aluminosilicate and vanadia could be possible; adsorption of the formed vanadium pentoxide on the surface of the fly ash; inclusion of vanadia in an aluminosilicate structure in either an interstitial, silicon or aluminium site.

Vanadium in Zirconia ($ZrSiO_4$) is believed to occupy tetrahedral Si^{4+} sites in the zirconia lattice.[6]

In the crystal structure of mullite [7] (Fig 4) the **aluminium atoms occupy octahedral sites and the silicon atoms tetrahedral sites**. The (010) and (001) planes indicate that interstitial positions are available for the vanadium atom.

Conclusion

The yellow colour resulting after fly ash reacted with ammonium metavanadate was believed to be the result of vanadium in the mullite structure itself, or the amorphous precursor of mullite.

Due to the similarity of the Raman spectrum of the yellow product and the Raman spectrum of vanadium pentoxide, it can be concluded that **the vanadium atoms possibly replaced some of the aluminium atoms in the mullite structure.**

References

1. Eskom - Lethabo Visitor Centre. Eskom, http://www.eskom.co.za/education/visit/lathabo/Lethabo_body.htm, 2002, downloaded 17 December 2002 - 15:30.
2. B.E. Scheetz, R. Earle, *Current Opinion in Solid State & Material Science*, 1998, **3**, 510-520.
3. L.D. Hulett, A.J. Weinberger, *Environmental Science and Technology*, 1980, 14(8), 965-970.
4. R. Helmuth, *Fly Ash in Cement and Concrete*, Portland Cement Association, Skokie, 1987, pp. 36-61.
5. J.L. Alonso, K. Wesche, *Characterization of Fly Ash*, in: K. Wesche (Ed), Fly Ash in Concrete - Properties and Performance, First edition, E & FN SPON, London, 1991, p. 8.
6. D. De Waal, A.M. Heyns, and others, *J. Raman Spectrosc.*, 1996, 27, 657-662.
7. R. Sadanga, M. Tokonami, Y. Takéuchi, *Acta Cryst.*, 1962, **15**, 65-68.

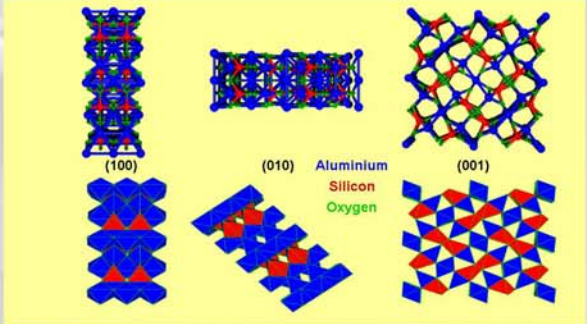
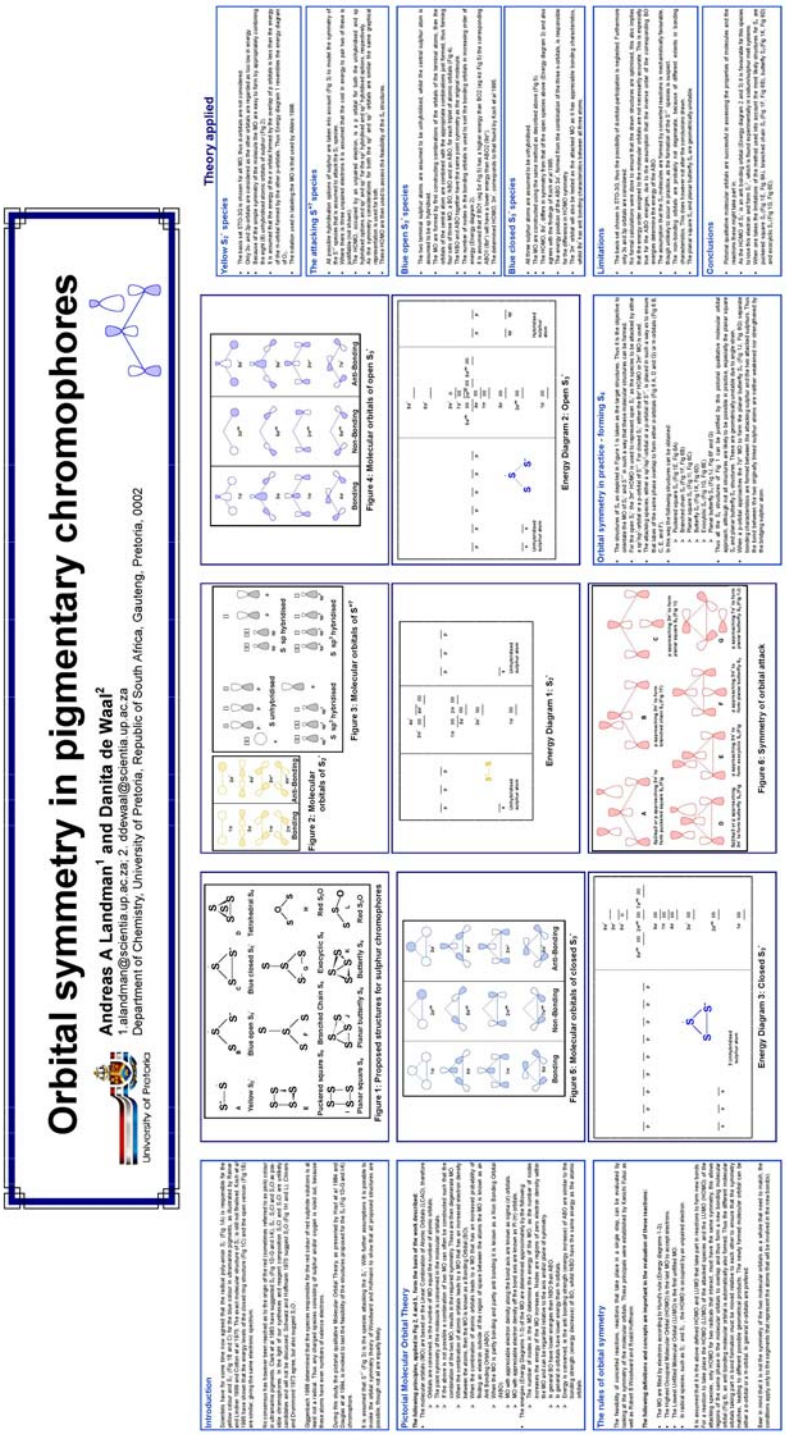


Figure 4: Crystal representations of Mullite

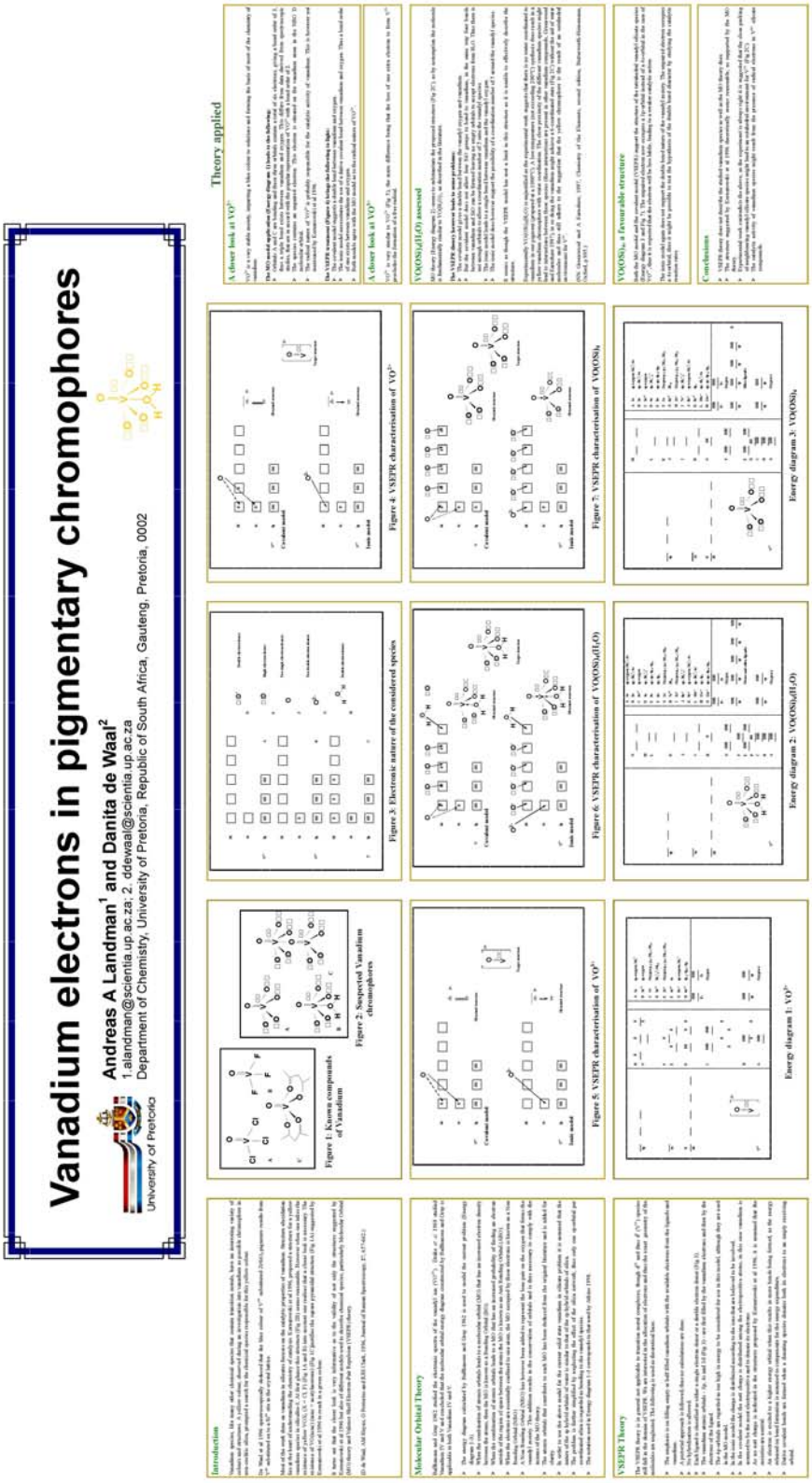
Acknowledgements

This work was supported by a grant from the Technology and Human Resources for Industry Programme (THRIP); a partnership programme funded by the Department of Trade and Industry (DTI) and managed by the National Research Foundation (NRF). The NRF, in a separate grant, also supported this work. The NRF in a separate grant also supported this work. The financial assistance of the Department of Labour (DoL) towards this research is hereby acknowledged. Opinions expressed and conclusions arrived at, are those of the author and are not necessarily to be attributed to the DoL. The authors also wish to thank Sphere-Fill (Pty) Ltd and the University of Pretoria for financial support.

Landman AA, De Waal D, *Symmetry in Pigmentary Chromophores*, proc INDABA III, p , 8 August 2000.



Landman AA, De Waal D, *Vanadium electrons in Pigmentary Chromophores*, proc INDABA III, p, 9 August 2000.



Vanadium electrons in pigmentary chromophores

Andreas A Landman[1] and Danita de Waal[2]
1.alandman@scientia.up.ac.za; 2. ddewaal@scientia.up.ac.za
Department of Chemistry, University of Pretoria, Republic of South Africa, Gauteng, Pretoria, 0002
University of Pretoria

Introduction

Molecular Orbital Theory

VSEPR Theory

Figure 1: Known compounds of Vanadium

Figure 2: Suspected Vanadium chromophores

Figure 3: Electronic nature of the considered species

Figure 4: VSEPR characterisation of VO^{2-}

Figure 5: VSEPR characterisation of VO^{3-}

Figure 6: VSEPR characterisation of $VOOSi_3(H_2O)$

Figure 7: VSEPR characterisation of $VOOSi_3$

Energy diagram 1: VO^{3-}

Energy diagram 2: $VOOSi_3(H_2O)$

Energy diagram 3: $VOOSi_3$

Theory applied

A closer look at VO^{3-}

A closer look at VO^{2-}

$VOOSi_3(H_2O)$ assessed

$VOOSi_3$, a favourable structure

Conclusions

De Waal D, Landman AA, Kruger RA, van Schalkwyk J, *Using Raman Spectroscopy to verify pigment identity*, proc Seventeenth International Conference on Raman Spectroscopy (ICORS 2000), Beijing, China, p , 20-24 August 2000.

Using Raman spectroscopy to verify pigment identity

D de Waal[1*], AA Landman[1], RA Kruger[2] and J van Schalkwyk[2]
[1]Department of Chemistry, University of Pretoria, Pretoria, 0002, Republic of South Africa;
E-Mails: ddewaal@scientia.up.ac.za , alandman@scientia.up.ac.za
[2]Spherefill, PO Box 3017, Randburg, 2125, Republic of South Africa

S_2^- S_3^-

Introduction

Bell, Clark and Gibbs set up a library of natural and synthetic pigments, published in 1997 [1]. One of the more interesting minerals discussed is Lazurite, a mineral (Lapis Lazuli) that has been synthetically copied as the industrial pigment Ultramarine Blue [2]. Ullmann's Encyclopedia [3] gives hints as to the industrial synthetic steps, but details of the synthesis of Ultramarine pigments are sketchy at best.

Although there were doubts as to the origin of the brilliant blue colour [4], it is now generally accepted that the blue chromophore is S_3^-, with a Raman peak at 549 cm^{-1} [4] (544±4 cm^{-1}). This species is accompanied by a yellow chromophore S_2^-, which is observed as a shoulder at about 589 cm^{-1} (588±5 cm^{-1}) in the Raman spectrum of the pigment. The origin of the red chromophore is still uncertain, but is probably an S_2 based chromophore [5].

The author has synthesised versions of Ultramarine Blue and Ultramarine Red. The synthesis follows what is believed to be a different route to those followed industrially, thus Raman spectroscopy with the aid of literature is used to identify the produced pigment as of Ultramarine type [5]. Preliminary results have shown that the new synthesis route leads to a stable product. Fourier Transform Infrared is used to establish that the necessary framework has been formed, to ensure the stability of the polysulphide [2, 6].

Reactions involved in Ultramarine chemistry

S_8 (Yellow) $\xrightarrow[\Delta]{\text{Reduction}}$ S_2^- (Yellow) $\xrightarrow[\Delta]{\text{Oxidation}}$ S_3^- (Blue) $\xrightarrow[\Delta]{\text{Oxidation}}$ S_4 (Red)

Pigment identification

The Raman bands of the synthesised products are compared to those of an obtained industrial Ultramarine Blue and Lazurite [1] (Table 1). From the spectra (Fig 1) it is clear that the synthesised product is very similar to the industrial Ultramarine Blue pigment. The main difference is the presence of Si-O bands in the Raman spectrum of the synthesised product.

The Raman spectrum of Ultramarine red (Fig 1) shows no bands linked to the S_3^- or S_2^- species, thus one can conclude that both sulphur species are converted to other chromophores. Although five bands can be observed in the Ultramarine red spectrum, only the band at 994 cm^{-1}, belonging to the Si-O stretching vibration, is identified. Further characterisation of the red chromophore is part of an ongoing study.

Following the oxidation of S_2^- to S_3^-

A partly synthesised Ultramarine Blue pigment batch was oxidised at a temperature of 1000 °C. Samples were taken at certain time intervals (Table 2) and the Raman spectra were later obtained under microscope (Fig 2, scaled to be comparable). The intensity ratio of the of the bands belonging to the S_3^- and S_2^- species were taken to indicate the progress of the oxidation of sulphur (Fig 3). The obtained spectra are baseline corrected and deconvoluted to obtain the intensities of the aforementioned bands.

As a trend was clear from all measurements except the measurements taken for the sample after 10 minutes (a rather short time) only two data points were chosen to represent this time period, referred to as the biased curve in Figure 3. The unbiased curve has been elevated for clarity. Although the system only follows the trend on average, as the uncertainties in the values are quite large, it is still valuable to note the gradual transition of the species.

Interpretation of the Raman spectra of Ultramarine pigments

- The yellow chromophore cannot be observed with the normal eye under the microscope, but electronic spectra confirms the yellow colour of the S_2^- species.
- The samples are inhomogeneous, ie both red and blue particles can be observed in the same sample, the amount ratio being a function of the oxidation time.
- Thus all results are biased in that they represent only the spectra of good examples of a certain species.
- Neither the starting reagent nor the blue particles show any significant bands belonging to the zeolite matrix, which encapsulates the chromophores. Thus the observed zeolite bands in the red particles can be taken to indicate that the matrix has been modified significantly.
- It is interesting to note that the Raman intensities for the blue species in the synthesised pigments are larger than those for the industrial pigments.

Conlusions

- Raman spectroscopy was used to verify the polysulphides as the chromophores in the prepared pigments (Fig 1, Table 1).
- Raman spectroscopy can be used to follow the oxidation reaction, during the formation and degradation of the blue pigment, S_3^- (Fig 3, Table 2).
- The spectrum of the red pigment indicates that the matrix that encapsulates the sulphur chromophore has changed, as the observed Si-O band is not observed in the spectra of the other pigments (Fig 1).
- Further work on the kinetics and structural properties of the ultramarine pigments is needed to fully understand the system.

Table 1: Raman bands[a] (cm^{-1}) of ultramarine pigments

Lazurite [1]	Ultramarine Red	Industrial Ultramarine Blue	Ultramarine Blue	Assignment
258 w b[b] Obs[c]	216 w b	255 w b 293 sh	257 w b 287 sh 323 sh	
	419 w b 461 w b			
548 st s Obs[c]		545 st s 568 sh	543 st s 573 sh	S_3^- S_2^-
822 w b	626 w b	813 w b	804 w b	
	994 st s		967 w b	Si- O
1096 m b		1097 m b	1092 m b	S_4

a) The following notation is used to describe the intensity and general shape of the peaks: w – weak, m – medium, st – strong, b – broad, sh – shoulder, s – sharp.
b) The description given differs from those in the cited reference, so as to have the same notation as the rest of the data.
c) The corresponding bands are observable, but have not been assigned, within the cited reference.

Table 2: Intensity ratio of S_2^- to S_3^-

Sample	S_3^- Peak	S_3^- Peak Intensity	S_2^- Peak	S_2^- Peak Intensity	Intensity Ratio	Average Ratio	Uncertainty
Starting Product	543.99 543.95 543.94 542.65	1908.58 3788.40 11462.20 3574.18	579.56 577.97 578.36 579.84	263.52 900.85 2454.75 853.64	0.17 0.24 0.21 0.24	0.22	0.06
After 10 minutes	543.78 542.73 541.79 540.67	1183.87 3456.38 1197.34 2283.56	578.87 578.99 578.44 575.00	215.09 670.65 483.31 958.82	0.18 0.19 0.39 0.42	0.19 0.30	0.03 0.27
After 30 minutes	546.42 545.01 544.17 542.48	593.68 705.42 1854.61 4272.75	585.87 582.17 580.27 574.35	21.96 82.00 287.99 857.25	0.11 0.12 0.16 0.20		
	543.63	1178.56	579.75	222.86	0.19	0.15	0.08
After 60 minutes	544.53 545.16 544.24 544.22	2868.56 2785.09 4318.10 6725.68	582.92 583.42 581.03 582.00	347.59 394.33 578.61 851.47	0.12 0.11 0.13 0.13	0.12	0.02
After 95 minutes	545.97 546.47 545.48 544.44	799.23 1253.29 1483.57 4829.83	578.03 583.10 582.38 578.32	68.20 93.69 157.15 617.86	0.09 0.07 0.11 0.13	0.10	0.05

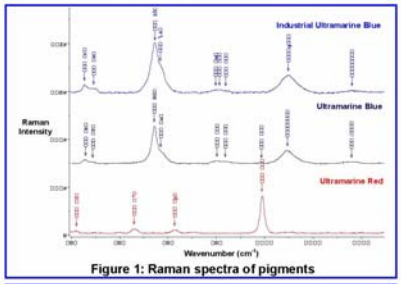


Figure 1: Raman spectra of pigments

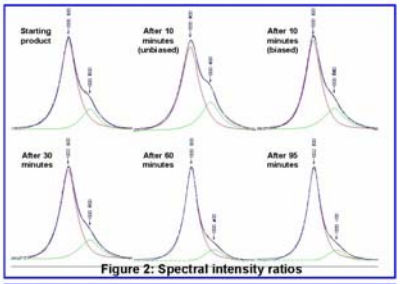


Figure 2: Spectral intensity ratios

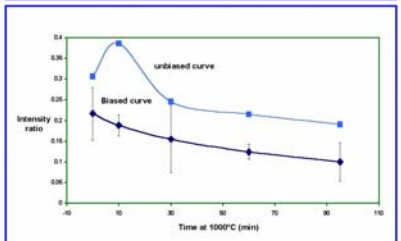


Figure 3: Intensity ratio of S_2^- to S_3^-

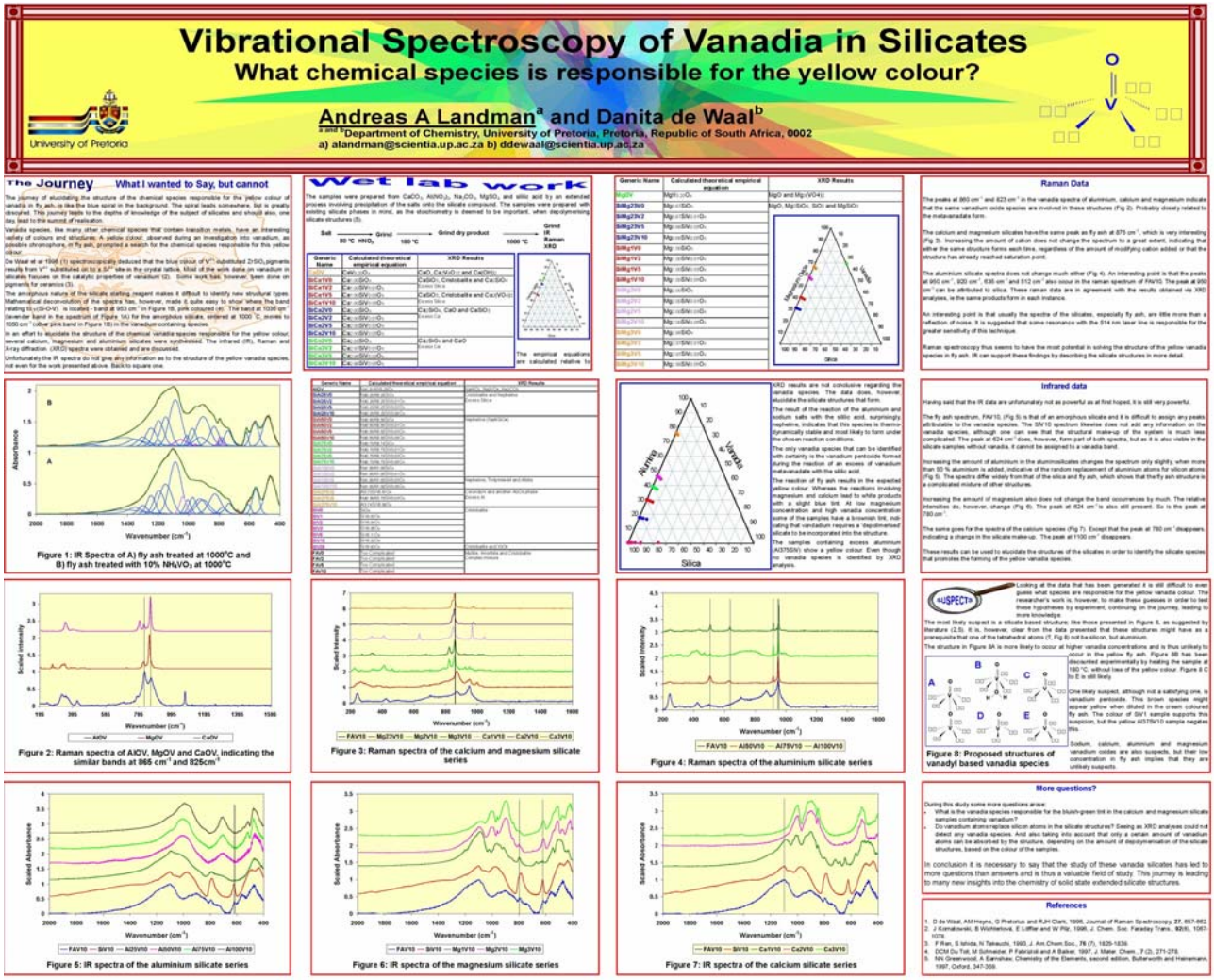
Acknowledgements

One of the authors gratefully acknowledges the financial support of
- The National Research Foundation of South Africa
- Spherefill
- The University of Pretoria

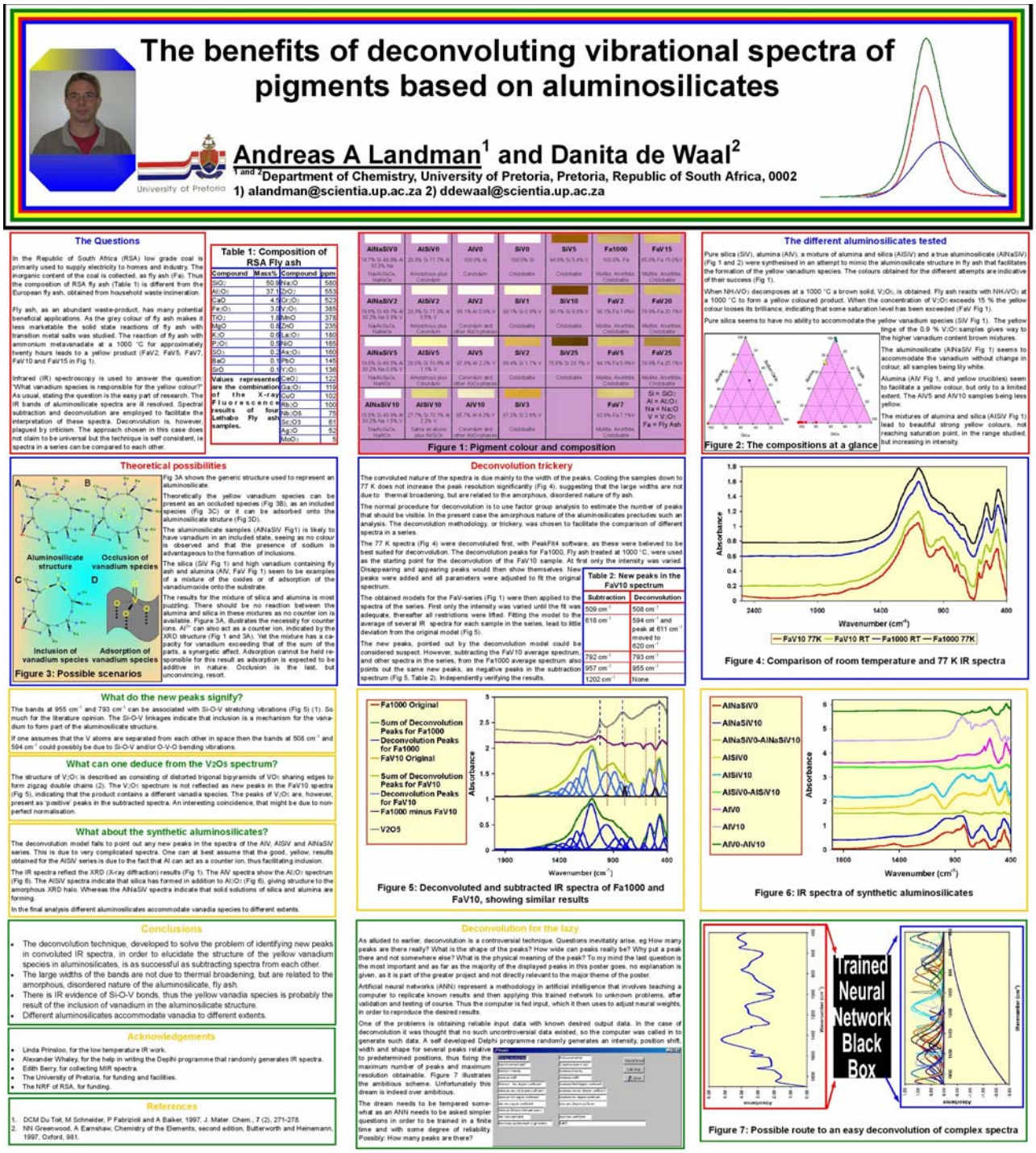
References

1. I.M. Bell, R.J.H. Clark, P.J. Gibbs, Raman spectroscopic library of natural and synthetic pigments (pre- ~1850AD), Spectrochimica Acta part A, 53, 2159, 1997
2. D Reinen, G Lindner, The nature of chalcogen colour centres in ultramarine-type solids, Chem.Soc.Rev. 28, 75, 1997
3. Ullmann's Encyclopedia of industrial chemistry, (B. Elvers, S. Hawkins, G. Schultz eds.), VCH Verlagsgesellschaft mbH, Weinheim, 1992, Vol A 20
4. U. Hofmann, E. Herzenstiel, K.-H. Schwarz, Ultramarin, Z. Anorg. Allg. Chemie, 367, 119, 1969
5. R.J.H. Clark, D.G. Cobbold, Characterization of Sulfur Radical Anions in Solutions of Alkali Polysulfides in Demethylformamide and Hexamethylphosphoramide and in the Solid State in Ultramarine Blue, Green, and Red, Inorganic Chemistry, 17, 3169, 1978
6. N. Gobeltz, A Demortier, J.P. Lelieur, Duhayon, Encapsulation of the chromophores into the sodalite structure during the synthesis of the blue ultramarine pigments, J.Chem.Soc. Faraday Trans., 94, 2257, 1998

Landman AA, De Waal D, *Vibrational Spectroscopy of Vanadia in Silicates*, proc Colloquium Spectroscopicum Internationale (CSI) XXXII, p P8-9, 10 July 2001.



Landman AA, De Waal D, *The Benefits of Deconvoluting Vibrational Spectra of Pigments Based on Aluminosilicates*, The First International Conference on Advanced Vibrational Spectroscopy ICAVS-1, p P5.44, 19 - 24 August 2001.



E. BENEFICIARIES

E1. SPHERE-FILL (PTY) LTD

"Sphere-Fill (Pty) Limited is a wholly owned subsidiary of Ash Resources, South Africa. Ash Resources currently sells over 1,2 million tons/annum of the finest quality fly ash in Southern Africa and the Middle East." (Contact Details - Company profile, Sphere-Fill, http://www.superpozz.com/contact1.html, 2002, downloaded 10 September 2002 - 10h00.) The spherical form of fly ash makes it potentially functional in improving the workability of plastics. The current grey colour of fly ash is however a deterrent, since this colour hides the colour of added pigments and fly ash is not marketable in its grey state. Sphere-Fill (Pty) Ltd needed to know whether it is possible to colour fly ash. The new colours observed in fly ash will make fly ash more marketable, especially the heat treated fly ash. Any company utilising the proposed pigment syntheses will be expected to sign a contract with Sphere-Fill (Pty) Ltd as exclusive supplier of fly ash.

E2. ROLFES COLOUR PIGMENTS INTERNATIONAL

Rolfes Colour Pigments International is interested in producing ultramarine blue locally. They can now consider using fly ash to produce the pigment.

E3. UNIVERSITY OF PRETORIA

"The University of Pretoria's mission is to:

- be an internationally recognised academic institution which provides teaching, undertakes research and renders community service;
- fulfil the educational, cultural, social, economic and technological needs of the South African and Southern African communities; and
- be a member of the international scientific community." (University of Pretoria Mission statement, University of Pretoria, http://www.up.ac.za/history/mission.html, 05 May 2001, downloaded 5 May 2001 - 22:30.)

This project aids in achieving these goals by:

- Research that is relevant to local companies
- Research output, with international exposure, in the form of journal articles (1 published, 1 accepted, 2 submitted and 4 prepared), conference contributions in the form of posters (9) and lectures (4) and a thesis/dissertation.
- Research grants help in extending and improving the facilities at the University of Pretoria.

E4. SCIENCE

Fly ash is a powerful starting reagent in the synthesis of ultramarine blue, and could possibly be used in other applications as well. Science gained from this research a methodology for studying the solid-state reactions of fly ash.

E5. ENVIRONMENT

Man harnesses energy for beneficial purposes. Unfortunately this often has a price. In the Republic of South Africa the price of using low grade, abundantly available coal in generating electricity is mountains of fly ash. The aim of this project is to find other uses for fly ash and therefore reduce the problem of unused resources that pose environmental problems. Less fly ash in the environment will benefit not only the Republic of South Africa but also all countries that use coal-fired power plants.

F. PUBLISHED ARTICLE