

ДОДАТОК А

Текст програми

```

using System;
using System.Linq;
using System.Drawing;
using System.Windows.Forms;
using MaterialSkin.Controls;
using TcPC.Types;
using TcPC.Localization;
namespace TcPC.Forms
{
    public partial class BurgerForm : MaterialForm, ILocalized
    {
        public BurgerForm()
        {
            InitializeComponent();
            ApplyLanguage();
            NumberInputFilter inputFilter = new NumberInputFilter();
            foreach (var numTextBox in new TextBox[]
            {
                _tcmParamValBox,
                _tcnParamValBox,
                _vnParamValBox,
                _cParamValBox
            }) numTextBox.KeyPress += new KeyPressEventHandler(inputFilter.OnKeyPress);
        }
        public void ApplyLanguage()
        {
            Text = LangResource.BurgerFormTitle;
            _inputTitle.Text = LangResource.InputTitle;
            _tcmParamDim.Text = LangResource.TcParamDim;
            _tcnParamDim.Text = LangResource.TcParamDim;
            _vnParamDim.Text = LangResource.VParamDim;
            _calculateButton.Text = LangResource.CalculateButtonText;
            _clearButton.Text = LangResource.ClearAllButtonText;
            _rangeSettings.StartTitle = LangResource.MinRangeParamTitle;
            _rangeSettings.EndTitle = LangResource.MaxRangeParamTitle;
            _rangeSettings.StepTitle = LangResource.StepParamTitle;
            _chart.ChartAreas[0].AxisX.Title = $"Vf, {LangResource.VParamDim}";
        }
    }
}

```

```

_chart.ChartAreas[0].AxisY.Title = $"λ, {LangResource.TcParamDim}";
_chart.Series[1].Name = LangResource.ExperimentalDataTitle;
_chart.Series[2].Name = LangResource.TheoreticalDataTitle;
_exportGraphPointsToolButton.Text = LangResource.ExportButtonText;
_outputHistoryTitle.Text = LangResource.OutputTitle;
_saveOutputHistoryButton.Text = LangResource.SaveButtonText;
_clearOutputHistoryButton.Text = LangResource.ClearButtonText;
_copyOutputHistoryButton.Text = LangResource.CopyButtonText;
_pointsInputTitle.Text = LangResource.ExperimentalDataTitle;
_pointsDataGrid.Columns[0].HeaderText = $"λ, {LangResource.TcParamDim}";
_buildPointsButton.Text = LangResource.BuildButtonText;
}
private void ClearButton_Click(object sender, EventArgs e)
{
    foreach (var textControl in new Control[]
    {
        _tcmParamValBox,
        _tcnParamValBox,
        _vnParamValBox,
        _cParamValBox
    }) textControl.Text = "0";
    _logger.Clear();
    _pointsDataGrid.Rows.Clear();
    foreach (var series in _chart.Series)
        series.Points.Clear();
    _chart.Visible = false;
}
private void BuildPointsButton_Click(object sender, EventArgs e)
{
    try
    {
        if (_chart.Series[2].Points.Count == 0)
        {
            _logger.Log(LangResource.NoDataAvailableComparisonMessageText, Color.Gold);
            return;
        }
        double vf = double.Parse(_vnParamValBox.Text);
        _chart.Series[1].Points.Clear();
        var rows = _pointsDataGrid.Rows;
        for (int i = 0; i < rows.Count - 1; i++)
        {
            double tce = Convert.ToDouble(rows[i].Cells[0].Value);
            _chart.Series[1].Points.AddXY(vf, tce);
        }
    }
}

```

```

        double tcff = _chart.Series[2].Points.First().YValues[0];
        double delta = Math.Abs(((tce * 100) / tcff) - 100);
        _logger.Log($"λ = {tce} {LangResource.TcParamDim} ({(int)delta} %)");
        if (vf < _rangeSettings.Start)
            _rangeSettings.Start = vf;
        if (vf > _rangeSettings.End)
            _rangeSettings.End = vf;
    }
    _chart.Visible = true;
}
catch (Exception ex)
{
    _logger.Log(ex.Message, Color.Red);
}
}

private double GetValTfVal(double tcm, double tcn, double Vf, double c) =>
    (((100 - Vf) * tcm) + (c * Vf * tcn)) / ((100 - Vf) + (c * Vf));
private void CalculateButton_Click(object sender, EventArgs e)
{
    try
    {
        double tcm = double.Parse(_tcmParamValBox.Text);
        double tcf = double.Parse(_tcnParamValBox.Text);
        double vf = double.Parse(_vnParamValBox.Text);
        double c = double.Parse(_cParamValBox.Text);
        double tc = GetValTfVal(tcm, tcf, vf, c);
        _logger.Log($"λm = { tcm } {LangResource.TcParamDim}");
        _logger.Log($"λf = { tcf } {LangResource.TcParamDim}");
        _logger.Log($"C = { c }");
        _logger.Log($"Vf = { vf } {LangResource.VParamDim}");
        _logger.Log($"λ = { Math.Round(tc, 3) } {LangResource.TcParamDim}\n");
        _chart.Series[0].Points.Clear();
        double start = _rangeSettings.Start;
        double end = _rangeSettings.End;
        if (vf >= start && vf <= end)
            _chart.Series[2].Points.AddXY(vf, tc);
        double step = (_rangeSettings.Step == 0) ? 1 : _rangeSettings.Step;
        for (vf = start; vf <= end; vf += step)
            _chart.Series[0].Points.AddXY(vf, GetValTfVal(tcm, tcf, vf, c));
        _chart.Visible = true;
    }
    catch (Exception ex)
    {

```

```

        _logger.Log(ex.Message, Color.Red);
    }
}
private void ExportChartPointButton_Click(object sender, EventArgs e)
{
    _saveFileDialog.FileName = "Points";
    if (_saveFileDialog.ShowDialog() != DialogResult.OK)
        return;
    PointExporter exporter = new PointExporter()
    {
        HorizontalAxisLabel = "Vf",
        VerticalAxisLabel = "λ"
    };
    exporter.Export(_saveFileDialog.FileName, _chart.Series[0].Points.ToArray());
    _logger.Log(LangResource.GraphExportMessageText, Color.LimeGreen);
}
private void LoggerClearContentButton_Click(object sender, EventArgs e) =>
    _logger.Clear();
private void LoggerSaveContentButton_Click(object sender, EventArgs e)
{
    _saveFileDialog.FileName = "Output";
    if (_saveFileDialog.ShowDialog() != DialogResult.OK)
        return;
    _logger.SaveContent(_saveFileDialog.FileName);
    _logger.Log(LangResource.OutputSavedMessageText, Color.LimeGreen);
}
private void LoggerCopyContentButton_Click(object sender, EventArgs e) =>
    _logger.CopyContent();
public partial class MainForm : MaterialForm, ILocalized
{
    private List<ILocalized> _localizedComponents;
    public MainForm()
    {
        InitializeComponent();
        ApplyVisualStyles();
        ApplyLanguage();
        _localizedComponents = new List<ILocalized>();
    }
    private void ApplyVisualStyles()
    {
        var skinManager = MaterialSkinManager.Instance;
        skinManager.Theme = MaterialSkinManager.Themes.DARK;
        skinManager.ColorScheme = new ColorScheme(Primary.Blue400,

```

```

        Primary.Blue800,
        Primary.BlueGrey500,
        Accent.LightBlue200,
        TextShade.WHITE);
    skinManager.AddFormToManage(this);
}
private void QuitButton_Click(object sender, EventArgs e)
{
    DialogResult dialogResult = MessageBox.Show(LangResource.QuitMessageBoxText,
        LangResource.QuitMessageBoxCaption,
        MessageBoxButtons.YesNo);
    if (dialogResult == DialogResult.Yes)
        Application.Exit();
}
private void GoTo(Form form)
{
    if (form is ILocalized)
    {
        ILocalized localizedComponent = (ILocalized)form;
        _localizedComponents.Add(localizedComponent);
        form.FormClosed += (sender, e) => _localizedComponents.Remove(localizedComponent);
    }
    form.Show();
}
private void ThermalConductivityFormButton_Click(object sender, EventArgs e)
{
    SelectModelForm selectModelForm = new SelectModelForm();
    selectModelForm.OnOk += GoTo;
    GoTo(selectModelForm);
}
private void PartCompositeFormButton_Click(object sender, EventArgs e) =>
    GoTo(new PartCompositeForm());
private void CoefficientCFormButton_Click(object sender, EventArgs e) =>
    GoTo(new CoefficientCForm());
private void PercolationThresholdFormButton_Click(object sender, EventArgs e) =>
    GoTo(new PercolationThresholdForm());
private void SetLanguage(string lang)
{
    Properties.Settings.Default.Language = lang;
    Properties.Settings.Default.Save();
    ApplyLanguage();
    foreach (ILocalized localizedComponent in _localizedComponents)
        localizedComponent.ApplyLanguage();
}

```

```

    }
    private void EnLangButton_Click(object sender, EventArgs e) => SetLanguage("en");
    private void RuLangButton_Click(object sender, EventArgs e) => SetLanguage("ru");
    private void UaLangButton_Click(object sender, EventArgs e) => SetLanguage("uk");
    public void ApplyLanguage()
    {
        string currentLanguage = Properties.Settings.Default.Language;
        CultureInfo currentCulture = new CultureInfo(currentLanguage, false);
        currentCulture.NumberFormat.NumberDecimalSeparator = ",";
        Thread.CurrentThread.CurrentUICulture = currentCulture;
        Thread.CurrentThread.CurrentCulture = currentCulture;
        _menuTitle.Text = LangResource.MenuTitle;
        _thermalConductivityFormButton.Text = LangResource.ThermalConductivityFormButtonText;
        _coefficientCFormButton.Text = LangResource.CoefficientCFormButton;
        _partCompositeFormButton.Text = LangResource.PartCompositeFormButtonText;
        _quitButton.Text = LangResource.QuitButtonText;
        _helpButton.Text = LangResource.HelpButtonText;
    }
    public partial class PartCompositeForm : MaterialForm, ILocalized
    {
        public PartCompositeForm()
        {
            InitializeComponent();
            ApplyLanguage();
            NumberInputFilter inputFilter = new NumberInputFilter();
            foreach (var numTextBox in new TextBox[]
            {
                _dmParamValBox,
                _vfParamValBox,
                _vmParamValBox,
                _widthParamValBox,
                _lengthParamValBox,
                _heightParamValBox,
            }) numTextBox.KeyPress += new KeyPressEventHandler(inputFilter.OnKeyPress);
        }
        public void ApplyLanguage()
        {
            Text = LangResource.PartCompositeFormTitle;
            _inputTitle.Text = LangResource.InputTitle;
            _dmParamDim.Text = LangResource.DParamDim;
            _vf2ParamDim.Text = LangResource.SizeDim;
            _vmParamDim.Text = LangResource.DParamDim;
            _vfParamDim.Text = LangResource.VParamDim;
        }
    }

```

```

_widthParamDim.Text = LangResource.VParamDim;
_heightParamDim.Text = LangResource.SizeDim;
_lengthParamDim.Text = LangResource.SizeDim;
_calculateButton.Text = LangResource.CalculateButtonText;
_clearButton.Text = LangResource.ClearAllButtonText;
_outputHistoryTitle.Text = LangResource.OutputTitle;
_saveOutputHistoryButton.Text = LangResource.SaveButtonText;
_clearOutputHistoryButton.Text = LangResource.ClearButtonText;
_copyOutputHistoryButton.Text = LangResource.CopyButtonText;
}
private void ClearButton_Click(object sender, EventArgs e)
{
    foreach (var textControl in new Control[]
    {
        _dmParamValBox,
        _vfParamValBox,
        _vmParamValBox,
        _widthParamValBox,
        _lengthParamValBox,
        _heightParamValBox,
    }) textControl.Text = "0";
    _logger.Clear();
}
private void CalculateButton_Click(object sender, EventArgs e)
{
    try
    {
        double widht = double.Parse(_heightParamValBox.Text);
        double height = double.Parse(_lengthParamValBox.Text);
        double lenght = double.Parse(_vf2ParamValBox.Text);
        double vs = widht * lenght * height;
        double dm = double.Parse(_dmParamValBox.Text);
        double v1 = double.Parse(_vmParamValBox.Text);
        double v2 = double.Parse(_vfParamValBox.Text);
        double v3 = double.Parse(_widthParamValBox.Text);
        double mn1 = vs * (v2 / 100) * dm;
        double mn2 = vs * (v3 / 100) * v1;
        double mm = (vs * ((100-(v2+v3))/100) * 1) / 0.125;
        double mk = mm + mn1 + mn2;
        _logger.Log($"w = { widht } {LangResource.SizeDim}");
        _logger.Log($"h = { height } {LangResource.SizeDim}");
        _logger.Log($"l = { lenght } {LangResource.SizeDim}");
        _logger.Log($"pf = { dm } {LangResource.DParamDim}");
    }
}

```

```

        _logger.Log($"Vm = { v1 } {LangResource.VParamDim}");
        _logger.Log($"Vf1 = { v2 } {LangResource.VParamDim}");
        _logger.Log($"Vf2 = { v3 } {LangResource.VParamDim}");
        _logger.Log($"Mf1 = { Math.Round(mn1, 3) } {LangResource.MDim}");
        _logger.Log($"Mf2 = { Math.Round(mn2, 3) } {LangResource.MDim}");
        _logger.Log($"Mm = { Math.Round(mm, 3) } {LangResource.MDim}");
        _logger.Log($"Mcom = { Math.Round(mk, 3) } {LangResource.MDim}\n"); ;
    }
    catch (Exception ex)
    {
        _logger.Log(ex.Message, Color.Red);
    }
}

private void LoggerClearContentButton_Click(object sender, EventArgs e) =>
    _logger.Clear();

private void LoggerSaveContentButton_Click(object sender, EventArgs e)
{
    _saveFileDialog.FileName = "Output";
    if (_saveFileDialog.ShowDialog() != DialogResult.OK)
        return;
    _logger.SaveContent(_saveFileDialog.FileName);
    _logger.Log(LangResource.OutputSavedMessageText, Color.LimeGreen);
}

private void LoggerCopyContentButton_Click(object sender, EventArgs e) =>
    _logger.CopyContent();

public partial class PercolationThresholdForm : MaterialForm, ILocalized
{
    public PercolationThresholdForm()
    {
        InitializeComponent();
        ApplyLanguage();
        NumberInputFilter inputFilter = new NumberInputFilter();
        foreach (var numTextBox in new TextBox[]
        {
            _tcmParamValBox,
            _tcnParamValBox,
            _vkParamValBox,
            _vnParamValBox,
            _vParamValBox,
            _tParamValBox,
            _sParamValBox
        }) numTextBox.KeyPress += new KeyPressEventHandler(inputFilter.OnKeyPress);
    }
}

```

```

public void ApplyLanguage()
{
    Text = LangResource.PercolationThresholdFormTitle;
    _inputTitle.Text = LangResource.InputTitle;
    _tcmParamDim.Text = LangResource.TcParamDim;
    _tcnParamDim.Text = LangResource.TcParamDim;
    _vkParamDim.Text = LangResource.VParamDim;
    _vnParamDim.Text = LangResource.VParamDim;
    _calculateButton.Text = LangResource.CalculateButtonText;
    _clearButton.Text = LangResource.ClearAllButtonText;
    _rangeSettings.StartTitle = LangResource.MinRangeParamTitle;
    _rangeSettings.EndTitle = LangResource.MaxRangeParamTitle;
    _rangeSettings.StepTitle = LangResource.StepParamTitle;
    _chart.ChartAreas[0].AxisX.Title = $"Vf, {LangResource.VParamDim}";
    _chart.ChartAreas[0].AxisY.Title = $"λ, {LangResource.TcParamDim}";
    _chart.Series[1].Name = LangResource.ExperimentalDataTitle;
    _chart.Series[2].Name = LangResource.TheoreticalDataTitle;
    _exportGraphPointsToolButton.Text = LangResource.ExportButtonText;
    _outputHistoryTitle.Text = LangResource.OutputTitle;
    _saveOutputHistoryButton.Text = LangResource.SaveButtonText;
    _clearOutputHistoryButton.Text = LangResource.ClearButtonText;
    _copyOutputHistoryButton.Text = LangResource.CopyButtonText;
    _pointsInputTitle.Text = LangResource.ExperimentalDataTitle;
    _pointsDataGrid.Columns[0].HeaderText = $"λ, {LangResource.TcParamDim}";
    _buildPointsButton.Text = LangResource.BuildButtonText;
}

private void ClearButton_Click(object sender, EventArgs e)
{
    foreach (var textControl in new Control[]
    {
        _tcmParamValBox,
        _tcnParamValBox,
        _vkParamValBox,
        _vnParamValBox,
        _vParamValBox,
        _tParamValBox,
        _sParamValBox
    }) textControl.Text = "0";
    _logger.Clear();
    _pointsDataGrid.Rows.Clear();
    foreach (var series in _chart.Series)
        series.Points.Clear();
    _chart.Visible = false;
}

```

```

}
private void BuildPointsButton_Click(object sender, EventArgs e)
{
    try
    {
        if (_chart.Series[2].Points.Count == 0)
        {
            _logger.Log(LangResource.NoDataAvailableComparisonMessageText, Color.Gold);
            return;
        }
        double vf = double.Parse(_vnParamValBox.Text);
        _chart.Series[1].Points.Clear();
        var rows = _pointsDataGrid.Rows;
        for (int i = 0; i < rows.Count - 1; i++)
        {
            double tce = Convert.ToDouble(rows[i].Cells[0].Value);
            _chart.Series[1].Points.AddXY(vf, tce);
            double tcff = _chart.Series[2].Points.First().YValues[0];
            double delta = Math.Abs(((tce * 100) / tcff) - 100);
            _logger.Log($"λ = {tce} {LangResource.TcParamDim} ({(int)delta} %)");
            if (vf < _rangeSettings.Start)
                _rangeSettings.Start = vf;
            if (vf > _rangeSettings.End)
                _rangeSettings.End = vf;
        }
        _chart.Visible = true;
    }
    catch (Exception ex)
    {
        _logger.Log(ex.Message, Color.Red);
    }
}
private double GetTcVal1(double tcm, double vk, double vf, double s) => tcm * Math.Pow(vk - vf, -s);
private double GetTcVal2(double tcn, double tcm, double v) => tcn * Math.Pow(tcm / tcn, v);
private double GetTcVal3(double tcn, double vf, double vk, double t) => tcn * Math.Pow((vf - vk), t);
private void CalculateButton_Click(object sender, EventArgs e)
{
    try
    {
        double vf1 = double.Parse(_vnParamValBox.Text);
        double vk1 = double.Parse(_vkParamValBox.Text);
        double tcm = double.Parse(_tcmParamValBox.Text);
        double tcf = double.Parse(_tcnParamValBox.Text);
    }
}

```

```

double v = double.Parse(_vParamValBox.Text);
double s = double.Parse(_sParamValBox.Text);
double t = double.Parse(_tParamValBox.Text);
double vk, vf;
vk = vk1 / 100;
vf = vf1 / 100;
double tc;
if (vf < vk)
{
    tc = GetTcVal1(tcm, vk, vf, s);
}
else if (vf == vk)
{
    tc = GetTcVal2(tcf, tcm, v);
}
else
{
    tc = GetTcVal3(tcf, vf, vk, t);
}
_logger.Log($"Vf = { vf } {LangResource.VParamDim}");
_logger.Log($"Vk = { vk } {LangResource.VParamDim}");
_logger.Log($"λm = { tcm } {LangResource.TcParamDim}");
_logger.Log($"λn = { tcf } {LangResource.TcParamDim}");
_logger.Log($"v = { v }");
_logger.Log($"s = { s }");
_logger.Log($"t = { t }");
_logger.Log($"λ = { Math.Round(tc, 3) } {LangResource.TcParamDim}\n");
foreach (var series in _chart.Series)
    series.Points.Clear();
_chart.Series[2].Points.AddXY(vf, tc);
double start = _rangeSettings.Start;
double end = _rangeSettings.End;
double step = (_rangeSettings.Step == 0) ? 0.1 : _rangeSettings.Step;
for (vf = start; vf <= end; vf += step)
{
    vf = Math.Round(vf, 2);
    vk = Math.Round(vk, 2);
    if (vf < vk)
    {
        tc = GetTcVal1(tcm, vk, vf, s);
    }
    else if (vf == vk)
    {

```

```

        tc = GetTcVal2(tcf, tcm, v);
    }
    else
    {
        tc = GetTcVal3(tcf, vf, vk, t);
    }
    _chart.Series[0].Points.AddXY(vf, tc);
}
_chart.Visible = true;
}
catch (Exception ex)
{
    _logger.Log(ex.Message, Color.Red);
}
}
private void ExportChartPointButton_Click(object sender, EventArgs e)
{
    _saveFileDialog.FileName = "Points";
    if (_saveFileDialog.ShowDialog() != DialogResult.OK)
        return;
    PointExporter exporter = new PointExporter()
    {
        HorizontalAxisLabel = "Vf",
        VerticalAxisLabel = "λ"
    };
    exporter.Export(_saveFileDialog.FileName, _chart.Series[0].Points.ToArray());
    _logger.Log(LangResource.GraphExportMessageText, Color.LimeGreen);
}
private void LoggerClearContentButton_Click(object sender, EventArgs e) =>
    _logger.Clear();
private void LoggerSaveContentButton_Click(object sender, EventArgs e)
{
    _saveFileDialog.FileName = "Output";
    if (_saveFileDialog.ShowDialog() != DialogResult.OK)
        return;
    _logger.SaveContent(_saveFileDialog.FileName);
    _logger.Log(LangResource.OutputSavedMessageText, Color.LimeGreen);
}
private void LoggerCopyContentButton_Click(object sender, EventArgs e) =>
    _logger.CopyContent();
public partial class SelectModelForm : MaterialForm, ILocalized
{
    public delegate void OkHandler(Form resultForm);

```

```

public event OkHandler OnOk;
public SelectModelForm()
{
    InitializeComponent();
    ApplyLanguage();
}
public void ApplyLanguage()
{
    _cancelButton.Text = LangResource.BackButtonText;
    _okButton.Text = LangResource.ContinueButtonText;
    _burgerRadioButton.Text = LangResource.BurgerModelCheckText;
    _percolationRadioButton.Text = LangResource.PercoltionModelCheckText;
}
private void PercolationRadioButton_CheckedChanged(object sender, EventArgs e)
{
    if(_percolationRadioButton.Checked)
        _burgerRadioButton.Checked = false;
}
private void BurgerRadioButtonCheckedChanged(object sender, EventArgs e)
{
    if (_burgerRadioButton.Checked)
        _percolationRadioButton.Checked = false;
}
private void CancelButtonClick(object sender, EventArgs e) =>
    Close();
private void OkButtonClick(object sender, EventArgs e)
{
    if (OnOk == null)
        return;
    Form resultForm;
    if (_burgerRadioButton.Checked)
        resultForm = new BurgerForm();
    else
        resultForm = new PercolationThresholdForm();
    OnOk.Invoke(resultForm);
    Close();
}
namespace TcPC.Forms
{
    public partial class CoefficientCForm : MaterialForm, ILocalized
    {
        public CoefficientCForm()
        {

```

```

InitializeComponent();
ApplyLanguage();
NumberInputFilter inputFilter = new NumberInputFilter();
foreach (var numTextBox in new TextBox[]
{
    _tcmParamValBox,
    _tcfParamValBox,
    _tcParamValBox,
    _vfParamValBox
}) numTextBox.KeyPress += new KeyPressEventHandler(inputFilter.OnKeyPress);
}
public void ApplyLanguage()
{
    Text = LangResource.CoefficientCFormTitle;
    _inputTitle.Text = LangResource.InputTitle;
    _tcmParamDim.Text = LangResource.TcParamDim;
    _tcnParamDim.Text = LangResource.TcParamDim;
    _tcfParamDim.Text = LangResource.TcParamDim;
    _vnParamDim.Text = LangResource.VParamDim;
    _calculateButton.Text = LangResource.CalculateButtonText;
    _clearButton.Text = LangResource.ClearAllButtonText;
    _chart.Series[1].Name = LangResource.ExperimentalDataTitle;
    _chart.Series[2].Name = LangResource.TheoreticalDataTitle;
    _rangeSettings.StartTitle = LangResource.MinRangeParamTitle;
    _rangeSettings.EndTitle = LangResource.MaxRangeParamTitle;
    _rangeSettings.StepTitle = LangResource.StepParamTitle;
    _exportGraphPointsToolButton.Text = LangResource.ExportButtonText;
    _outputHistoryTitle.Text = LangResource.OutputTitle;
    _saveOutputHistoryButton.Text = LangResource.SaveButtonText;
    _clearOutputHistoryButton.Text = LangResource.ClearButtonText;
    _copyOutputHistoryButton.Text = LangResource.CopyButtonText;
    _pointsInputTitle.Text = LangResource.ExperimentalDataTitle;
    _buildPointsButton.Text = LangResource.BuildButtonText;
}
private void ClearButton_Click(object sender, EventArgs e)
{
    foreach (var textControl in new Control[]
    {
        _tcmParamValBox,
        _tcfParamValBox,
        _vfParamValBox,
        _tcParamValBox
    }) textControl.Text = "0";
}

```

```

_logger.Clear();
_pointsDataGrid.Rows.Clear();
foreach (var series in _chart.Series)
    series.Points.Clear();
_chart.Visible = false;
}
private void BuildPointsButton_Click(object sender, EventArgs e)
{
    try
    {
        if (_chart.Series[2].Points.Count == 0)
        {
            _logger.Log(LangResource.NoDataAvailableComparisonMessageText, Color.Gold);
            return;
        }
        double tcm = double.Parse(_tcmParamValBox.Text);
        double tcf = double.Parse(_tcfParamValBox.Text);
        double tc = tcf / tcm;
        _chart.Series[1].Points.Clear();
        var rows = _pointsDataGrid.Rows;
        for (int i = 0; i < rows.Count - 1; i++)
        {
            double c = Convert.ToDouble(rows[i].Cells[0].Value);
            _chart.Series[1].Points.AddXY(tc, c);
            double cT = _chart.Series[2].Points.First().YValues[0];
            double delta = Math.Abs(((c * 100) / cT) - 100);
            _logger.Log($"C = {c} {LangResource.TcParamDim} ({(int)delta} %)");
            if (tc < _rangeSettings.Start)
                _rangeSettings.Start = tc;
            if (tc > _rangeSettings.End)
                _rangeSettings.End = tc;
        }
        _chart.Visible = true;
    }
    catch (Exception ex)
    {
        _logger.Log(ex.Message, Color.Red);
    }
}
private double GetCVal(double tcm, double tc, double tcf, double vf) =>
    -((tc * (100 - vf)) - ((100 - vf) * tcm)) / ((vf * tc) - (vf * tcf));
private double GetC1Val(double tcm, double tc, double tcf, double vf) =>
    -((tc * (100 - vf)) - ((100 - vf) * tcm)) / ((vf * tc) - (vf * (tcf / tcm)));

```

```

private void CalculateButton_Click(object sender, EventArgs e)
{
    try
    {
        double tcm = double.Parse(_tcmParamValBox.Text);
        double tcf = double.Parse(_tcfParamValBox.Text);
        double tc = double.Parse(_tcParamValBox.Text);
        double vf = double.Parse(_vfParamValBox.Text);
        double c = GetCVal(tcm, tc, tcf, vf);
        double c1 = GetC1Val(tcm, tc, tcf, vf);
        double tcmX = tcf / tcm;
        _logger.Log($"λm = {tcm} {LangResource.TcParamDim}");
        _logger.Log($"λf = {tcf} {LangResource.TcParamDim}");
        _logger.Log($"λ = {tc} {LangResource.TcParamDim}");
        _logger.Log($"Vf = {vf} {LangResource.VParamDim}");
        _logger.Log($"C = {Math.Round(c, 3)}");
        // _logger.Log($"C(λf/λm) = {Math.Round(c1, 3)}");
        _logger.Log($"λf/λm = {Math.Round(tcmX, 3)}\n");
        foreach(var series in _chart.Series)
            series.Points.Clear();
        double start = _rangeSettings.Start;
        double end = _rangeSettings.End;
        double step = (_rangeSettings.Step == 0) ? 1 : _rangeSettings.Step;
        if (tcmX >= start && tcmX <= end)
            _chart.Series[2].Points.AddXY(tcmX, c1);
        tcf = start;
        double tcVal = tcf / tcm;
        while(tcVal <= end)
        {
            _chart.Series[0].Points.AddXY(tcVal, GetC1Val(tcm, tc, tcf, vf));
            tcf += step;
            tcVal = tcf / tcm;
        }
        _chart.Visible = true;
    }
    catch (Exception ex)
    {
        _logger.Log(ex.Message, Color.Red);
    }
}

private void ExportChartPointButton_Click(object sender, EventArgs e)
{
    _saveFileDialog.FileName = "Points";
}

```

```
if (_saveFileDialog.ShowDialog() != DialogResult.OK)
    return;
PointExporter exporter = new PointExporter()
{
    HorizontalAxisLabel = " $\lambda f / \lambda m$ ",
    VerticalAxisLabel = "C"
};
exporter.Export(_saveFileDialog.FileName, _chart.Series[0].Points.ToArray());
_logger.Log(LangResource.GraphExportMessageText, Color.LimeGreen);
}
private void LoggerClearContentButton_Click(object sender, EventArgs e) =>
    _logger.Clear();
private void LoggerSaveContentButton_Click(object sender, EventArgs e)
{
    _saveFileDialog.FileName = "Output";
    if (_saveFileDialog.ShowDialog() != DialogResult.OK)
        return;
    _logger.SaveContent(_saveFileDialog.FileName);
    _logger.Log(LangResource.OutputSavedMessageText, Color.LimeGreen);
}
```

ДОДАТОК Б

Публікація до кваліфікаційної роботи



ОСОБЛИВОСТІ ОТРИМАННЯ ОДНОРАЗОВОЇ ГРОШОВОЇ ДОПОМОГИ ПОВНОЛІТНІМИ ДІТЬМИ ЗАГИБЛИХ ВІЙСЬКОВОСЛУЖБОВЦІВ Вареник Д.С., Бингар Я.Ю., Куйдан О.В., Ухань І.К., Науковий керівник: Максимова М.К.	72
ОСОБЛИВОСТІ ПРАВОНАСТУПНИЦТВА ДЕРЖАВ ЩОДО ВІДПОВІДАЛЬНОСТІ ЗА МІЖНАРОДНО-ПРОТИПРАВНІ ДІЯННЯ ЗГІДНО ПОЗИЦІЙ КОМІСІЇ ООН З МІЖНАРОДНОГО ПРАВА Медведенко І.Ю.	77
ОСОБЛИВОСТІ СИСТЕМИ МІЖНАРОДНО-ПРАВОВИХ АКТИВ ЩОДО БОРОТЬБИ З КАТУВАННЯМ Шолох В.А.	86

РОЗДІЛ 6. ІНСТИТУТ ПРАВООХОРОННОЇ ДІЯЛЬНОСТІ, СУДОВА СИСТЕМА ТА НОТАРІАТ

ОБСТАВИНИ, ЩО ВИКЛЮЧАЮТЬ КРИМІНАЛЬНУ ПРОТИПРАВНІСТЬ ДІЯННЯ Чернявська О.О., Науковий керівник: Бабанін С.В.	91
---	----

РОЗДІЛ 7. ПОЖЕЖНА ТА ЦИВІЛЬНА БЕЗПЕКА

ВИМОГИ ДО ЯКОСТІ ВОДИ ДЛЯ ЛІКЕРО-ГОРІЛЧАНИХ НАПОЇВ Павлюк Ю.В., Шліхта О.О., Науковий керівник: Бабаджанова О.Ф.	94
НЕБЕЗПЕКА АЗБЕСТУ ТА ВПЛИВ НА ОРГАНІЗМ ЛЮДИНИ Гончаренко М.О., Татусь А.С., Радзіонов К.С., Науковий керівник: Синельников О.Д.	100
ОСОБЛИВОСТІ ВПЛИВУ ДЖЕРЕЛ РАДІОАКТИВНОГО ЗАБРУДНЕННЯ ТА ПРИНЦИПИ РАДІАЦІЙНОГО ЗАХИСТУ НАСЕЛЕННЯ Гончаренко М.О., Мартинов І.М., Михалік Н.П., Науковий керівник: Лоїк В.Б.	107

РОЗДІЛ 8. АВТОМАТИЗАЦІЯ ТА ПРИЛАДОБУДУВАННЯ

РОЗРОБКА ПРОГРАМИ ДЛЯ АВТОМАТИЗАЦІЇ ПРОЦЕСУ РОЗРАХУНКУ ЕФЕКТИВНОЇ ТЕПЛОПРОВІДНОСТІ НОВИХ ПОЛІІМІДНИХ КОМПОЗИТНИХ МАТЕРІАЛІВ Борщов І.В., Науковий керівник: Новоселов С.П.	114
--	-----

РОЗДІЛ 9. ЕЛЕКТРОНІКА ТА ТЕЛЕКОМУНІКАЦІЇ

ОГЛЯД МЕТОДІВ ГЕНЕРАЦІЇ ЛІНІЙНИХ ПСЕВДОВИПАДКОВИХ ПОСЛІДОВНОСТЕЙ ДЛЯ ПСЕВДОВИПАДКОВОГО ПЕРЕНАЛАШТУВАННЯ РОБОЧОЇ ЧАСТОТИ Меркелов І.В.	127
---	-----

Борщов І.В.

здобувач вищої освіти факультету автоматики та комп'ютеризованих технологій
Харківський національний університет радіоелектроніки, Україна

Науковий керівник: Новоселов С.П.

канд. техн. наук, доцент, професор кафедри комп'ютерно-інтегрованих
технологій, автоматизації та робототехніки
Харківський національний університет радіоелектроніки, Україна

РОЗРОБКА ПРОГРАМИ ДЛЯ АВТОМАТИЗАЦІЇ ПРОЦЕСУ РОЗРАХУНКУ ЕФЕКТИВНОЇ ТЕПЛОПРОВІДНОСТІ НОВИХ ПОЛІІМІДНИХ КОМПОЗИТНИХ МАТЕРІАЛІВ

АНОТАЦІЯ. В роботі розглянуто теоретичні аспекти теплопровідності поліімідних композитів, враховуючи моделі Бургера та перколяції. За допомогою програмного комплексу COMSOL MULTIPHYSICS виконано 3D-структурне моделювання комірок поліімідних композитних плівок. Виконані розрахунки теплопровідності, зокрема по формулі Бургера та моделі перколяції. Розглянуті їхні переваги та недоліки. Розроблено та написано спеціалізовану програму «*Теплопровідність поліімідного композиту TcPC*», яка дозволяє автоматизувати процеси розрахунку коефіцієнтів теплопровідності для поліімідних композитних матеріалів за аналітичною моделлю Бургера та моделлю перколяції.

КЛЮЧОВІ СЛОВА: *теплопровідність, поліімідні композити, моделювання, програма розрахунку, ефективний коефіцієнт теплопровідності.*

ВСТУП

При використанні поліімідних композитних матеріалів у системах електричної ізоляції та комутуючих елементів електронних вузлів актуальним

РОЗДІЛ 8.**АВТОМАТИЗАЦІЯ ТА ПРИЛАДОБУДУВАННЯ**

є завдання покращення теплопередачі з одночасним збереженням високих діелектричних характеристик теплопровідних поліімідних шарів.

Прогнозування ефективної теплопровідності наповнених поліімідних систем вимагає знання не тільки теплових властивостей складових компонентів, але й ряд інших факторів. Дуже важливим з цієї точки зору є розробка моделі наповненого поліімиду, а також моделювання граничних шарів, які дуже впливають на властивості наповненої композитної системи.

Необхідно враховувати, що механізми впливу матриці та наповнювача на теплопровідність різні за різних концентрацій. Управління властивостями поліімідних композитних матеріалів має забезпечуватися не тільки збільшенням концентрації, а також збільшенням модифікуючої дії наповнювача (за рахунок дисперсності та інших факторів).

Для наближених розрахунків ефективної теплопровідності двокомпонентних сумішей для широкого діапазону вмісту порошкових наповнювачів у композитах використовувати прості лінійні моделі, наприклад формулу Бургера та модель перколяції. Для підвищення ефективності таких розрахунків необхідна розробка спеціалізованих програм, які дозволять автоматизувати і прискорити процес отримання оціночних приблизних даних про коефіцієнти теплопровідності композиційних матеріалів, що розробляються.

Використання чисельних методів для знаходження ефективної теплопровідності композиційної системи дозволяє проводити більш точні розрахунки для різних розподілів наповнювача в матриці, що дозволяє врахувати вплив структури на теплові властивості композиту, а також наявність граничного шару.

Таким чином метою статті є програма «Теплопровідність поліімідного композиту» за аналітичною моделлю Бургера та моделлю перколяції.

1. Предмет та методи дослідження

1.1. Структурне моделювання для розрахунку ефективної теплопровідності теплопровідних поліімідних композитних плівок за аналітичною моделлю Бургера та моделлю перколяції

1.1.1. Структурне моделювання теплопровідних поліімідних композитів

За допомогою програмного комплексу COMSOL MULTIPHYSICS були виконані 3D-структурне моделювання комірок поліімідних композитних плівок.

Теоретичний опис явищ перенесення (електропровідність, теплопровідність та ін.) у твердих тілах, рідких та газоподібних середовищах заснований на схожості основоположних рівнянь. Саме подібність рівнянь позначилося на теорії «узагальненої провідності». "Узагальнена провідність" поєднує властивості композитів, чутливих до їх структури. Такими властивостями є: діелектрична та магнітна проникність, теплопровідність,

модуль зсуву, пружність тощо.

Значення властивостей теплопровідних композитів залежить як від об'ємної частки компонентів у ПКМ, так і від структури ПКМ та орієнтації меж розділу сполучного та наповнювача.

Аналізуючи теоретичні та експериментальні дослідження сумішей та ПКМ [1, 2, 3, 4], що містять більше однієї фази, можна зробити ряд основних допущень про властивості та характеристики функціональних матеріалів.

До них відносяться:

- геометрія твердих частинок - проста і правильна;
- заданий розподіл частинок наповнювача у сполучному (хаотичний/упорядкований). Це дозволяє виділити типову елементарну комірку, що характеризує матеріал взагалі;
- взаємодія фаз наповнювача та/або сполучного (ізолюваність або зв'язність);
- тепловий контакт частинок наповнювача (ідеальний чи обумовлений);
- наявність/відсутність хімічних взаємодій між наповнювачем та сполучним;
- перенесення тепла розглядається за основним механізмом теплопровідності.

Завданням структурного моделювання є розробка поліімідних композитних матеріалів, у яких можна використовувати дрібнодисперсні наповнювачі - аж до нанорозмірних частинок - і забезпечити рівномірний розподіл теплопровідного наповнювача в поліімідній матриці, що гарантує високий рівень теплопровідних, діелектричних і фізико-механічних властивостей одержуваного функціонального матеріалу.

Ще одним завданням структурного моделювання є також отримання теплопровідних електроізолюючих поліімідних композитних матеріалів (варіантів), що володіють високими теплопровідними і діелектричними властивостями за умови збереження хороших фізико-механічних властивостей навіть за надвисоких ступенів наповнення.

В роботі [6, 7] повідомляється, що зменшення розміру частинок наповнювачів у композитному матеріалі до нанорозмірів не призводить до істотної зміни теплопровідності порівняно з матеріалами, що використовують мікронні частинки наповнювача. Зменшення розміру наночастинок менше, ніж 20 нм призводить до значного зниження значення коефіцієнта теплопровідності. При використанні мікронних і нанорозмірних частинок у суміші теплопровідність композитів збільшується. При цьому розмір мікро- і наночастинок наповнювачів у використовуваній суміші впливає на властивості композиту не так помітно, як їхнє масове співвідношення в суміші. Відношення розміру мікрочастинок до розміру d наночастинок не повинно

РОЗДІЛ 8.**АВТОМАТИЗАЦІЯ ТА ПРИЛАДОБУДУВАННЯ**

перевищувати 1000, але й не повинно бути меншим за 100, тому що інакше не досягається оптимальний розподіл наночастинок між мікрочастинками та не утворюється достатньої кількості теплопровідних шляхів, унаслідок чого теплопровідність композиційного матеріалу падає.

На рис. 1.1 наведено структурна схема ПІ композиту з відношенням розміру мікронних частинок до розміру наночастинок при якому теплопровідність композиту може збільшуватись за рахунок оптимального розподілу наночастинок між мікронними частинками при достатній кількості теплопровідних шляхів у композиті.

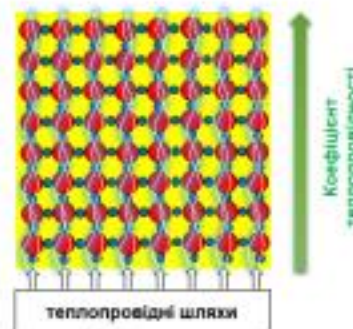


Рис. 1.1. Структурна схема ПІ композиту з оптимальним розподілом наночастинок між мікронними частинками при відношенні розміру мікрочастинок до розміру наночастинок $100 < d < 1000$

Високі теплопровідні властивості досягаються за наявності часток наповнювача в композиті не менше ніж 55 мас. % і зростають за подальшого збільшення вмісту наповнювача (рис. 1.2а). Збільшення вмісту наповнювача понад 90 мас. % призводить до погіршення міцності та пластичності композитних ПІ плівок (рис. 1.2б).

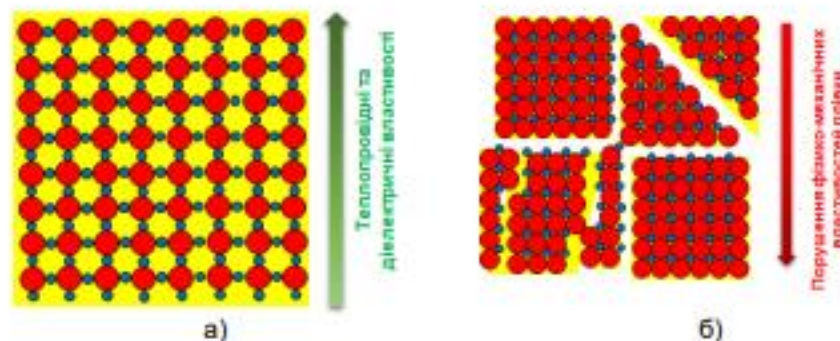


Рис. 1.2. Структурна схема ПІ композиту: а) вміст наповнювача не менше 55 мас. %, б) вміст наповнювача більше 90 мас. %

1.1.2. Розрахункова модель теплопровідності КМ за формулою Бургера

З погляду прогнозування такого найважливішого показника якості, як теплопровідність КМ, важливо переконатися в адекватності різних моделей на сукупності експериментальних даних, що охоплюють такі основні параметри:

- матеріал наповнювача;
- матеріал сполучного;
- об'ємна частка наповнювача в композитному матеріалі.

Для наближених оцінок ефективної теплопровідності двох та трьох компонентних сумішей для широкого діапазону вмісту порошкових наповнювачів показано можливість застосування лінійних моделей [3, 4, 5]. Це дозволяє для практичних оцінок використовувати прості розрахункові моделі, наприклад формулу Бургера (1.1) для сферичних частинок. Такий підхід забезпечує високу точність результатів, зберігаючи при цьому достатню простоту і зручність в застосуванні.

Під визначенням «сполучна» розуміється безперервна фаза, а під «наповнювачем» — уривчаста. Наявність коефіцієнта C у формулі Бургера робить її зручною для опису теплопровідності двокомпонентних композитних матеріалів. Величина цього коефіцієнта враховує такі характеристики матеріалу, як безперервність/ уривчастість фаз кожного з компонентів, форму і розмір частинок наповнювача, і відношення їх коефіцієнтів теплопровідності. Саме в цьому полягає його фізичне значення. Безрозмірний коефіцієнт C залежить, по-перше, від характеристик розподілу фаз у суміші: $C = 1$ – гомогенна суміш, $C \ll 1$ – ізольована фаза наповнювача, $C > 1$ – безперервна фаза наповнювача. По-друге, від відношення λ_w / λ_{ce} (чим більше це відношення, тим менший коефіцієнт C) (рис. 1.3).

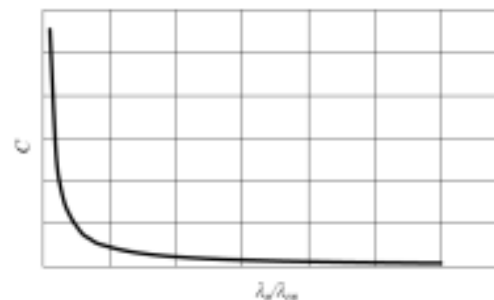


Рис. 1.3. Залежність коефіцієнта C у формулі Бургера від величини відношення λ_w / λ_{ce}

Для розрахунку коефіцієнта C необхідно виготовити та дослідити

РОЗДІЛ 8.**АВТОМАТИЗАЦІЯ ТА ПРИЛАДОБУДУВАННЯ**

ефективну теплопровідність експериментального зразка поліімідних композитних плівок і розрахувати за формулою:

$$C = - \frac{\lambda_{эф} V_{ce} - V_{ce} \lambda_{ce}}{V_n \lambda_{эф} - V_n \lambda_n} \quad (1.1)$$

Недоліком формули Бургера є необхідність підбору значення коефіцієнта "С" для кожного конкретного КМ, що знижує її універсальність. Також ця модель не враховує перколяційну складову теплопровідності.

1.1.3. Розрахункова модель теплопровідності КМ за моделлю перколяції (порога протікання)

Згідно з теорією перколяції, поблизу порога протікання частинки наповнювача утворюють стійкі кластери, у поверхневих шарах яких швидкості процесів перенесення вищі, ніж в об'ємі матеріалу. Унаслідок цього зближення кластерів за критичного об'єму наповнювача й утворення ними протяжної структури (так званий "нескінченний" кластер) супроводжуються швидким посиленням явищ переносу в двофазному матеріалі. Точка перколяції відповідає концентрації наповнювача, за якої розмір зв'язного кластера досягає розміру системи. Подальше збільшення концентрації наповнювача і відхилення фази включень від стійкої кластерної структури супроводжуються зменшенням ролі меж кластерів у явищах переносу. Таким чином, виникнення особливих геометрично стійких утворень зі специфічними властивостями аналогічно утворенню структур з фрактальними характеристиками у неоднорідних невпорядкованих середовищах. [5].

У деяких моделях опису властивостей композиційних матеріалів поблизу порога протікання враховується стрибкоподібна зміна властивостей (електропровідність, теплопровідність тощо). Межа зміни властивостей реальної системи розмивається і проявляється як область особливої поведінки залежно від таких факторів:

- рівень неоднорідності властивостей сполучного і наповнювача;
- форма і розмір частинок і структури фази наповнювача;
- властивості компонентів (вільна енергія, поверхневий натяг, змочуваність, в'язкість, адгезія);
- ступінь агломерації частинок наповнювача при введенні його в матрицю;
- співвідношення граничної концентрації наповнювача в сполучному і концентрації, що відповідає порогу протікання;
- технологія приготування зразка (змішування, спікання, термообробка, пресування тощо).

Коефіцієнт теплопровідності двофазного композитного матеріалу за

теорією перколяції описується такою системою [4]:

$$\begin{aligned} \lambda_{\text{эф}} &= \lambda_{\text{се}} (V_{\text{кр}} - V_n)^{-s} \text{ при } V_n < V_{\text{кр}} \\ \lambda_{\text{эф}} &= \lambda_n \left(\frac{\lambda_{\text{се}}}{\lambda_n} \right)^v \text{ при } V_n = V_{\text{кр}} \\ \lambda_{\text{эф}} &= \lambda_n (V_n - V_{\text{кр}})^t \text{ при } V_n > V_{\text{кр}} \end{aligned} \quad (1.2)$$

де $\lambda_{\text{се}}$ - коефіцієнт теплопровідності сполучного;
 λ_n - коефіцієнт теплопровідності наповнювача;
 $V_{\text{кр}}$ - критичний об'єм наповнювача;
 V_n - об'єм наповнювача;
 s, v, t - критичний індекс.

Значення критичних індексів підбираються так, щоб забезпечити

- найкраща згода розрахункових та експериментальних величин теплопровідності за умов $V_n < V_{\text{кр}}$;
- монотонне зростання теплопровідності зі збільшенням вмісту наповнювача поблизу точки $V_{\text{кр}}$;
- співвідношення $\lambda_{\text{эф}}/\lambda_n \approx V^2$ за $V_n > V_{\text{кр}}$, що впливає з розрахункової моделі теплопровідності двофазних гетерогенних матеріалів, у яких частинки наповнювача розподілені в об'ємі матриці випадковим чином.

Недоліком моделі перколяції є необхідність підбору значень критичний індексів для кожного конкретного КМ через експерименті дані ефективної теплопровідності композитного матеріалу.

2. Результати та їх обговорення

2.1. Розробка та написання програми «Теплопровідність поліімідного композиту ТсРС»

Для ефективної автоматизації розрахунків теплопровідності поліімідних композитних плівок за моделями Бургера та перколяції була створена програма «Теплопровідність поліімідного композиту ТсРС». Розробка програми виконана на платформі NET framework і використовує мову програмування C# для забезпечення зручного та ефективного використання.

Основні функції та можливості програми.

Розрахунок ефективного коефіцієнта теплопровідності:

- програма дозволяє виконати розрахунки ефективного коефіцієнта теплопровідності композитної плівки з використанням моделей Бургера та перколяції. Результати розрахунків можна відобразити у вигляді графіків, що дозволяє візуалізувати залежність теплопровідності від об'ємної концентрації наповнювача.

РОЗДІЛ 8.**АВТОМАТИЗАЦІЯ ТА ПРИЛАДОБУДУВАННЯ**

Розрахунок та вибір коефіцієнта «С»:

– програма виконує розрахунок та вибір коефіцієнта «С» за експериментальними даними. Результати представлені у вигляді графіків, що вказує на залежність коефіцієнта «С» від відношення теплопровідностей.

Розрахунок концентрації компонентів:

– програма дозволяє розраховувати об'ємну і масову концентрацію компонентів, необхідних для виготовлення композиту.

Порівняння експериментальних та теоретичних даних:

– підтримка порівняння експериментальних та теоретичних даних, що дозволяє оцінити відхилення між ними. Результати порівняння виводяться у зручному для аналізу вигляді.

Представлення інформації мовами:

– програма надає можливість представлення інформації як українською, так і англійською мовами, що робить її зручною та доступною для широкого кола користувачів.

Програма «Теплопровідність поліімідного композиту TcPC» є потужним інструментом для вчених та дослідників у галузі матеріалознавства, який дозволяє виконувати точні та ефективні розрахунки теплопровідності поліімідних композитних плівок з використанням сучасних аналітичних моделей.

Інтерфейс користувача

Інтерфейс користувача створено за допомогою інструменту *Windows Forms*. Основне вікно програми показано на рис. 2.1.

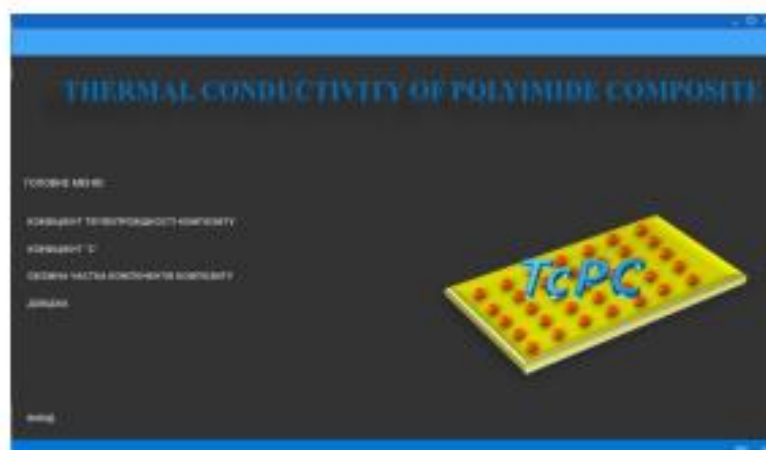


Рис. 2.1. Головне меню програми TcPC

На головному меню знаходяться основні активні кнопки переходу між вікнами програми, зміни мови інтерфейсу, довідка та вихід з програми.

Перехід здійснюється за допомогою методу GoTo.

Кнопка *ДОВІДКА* дозволяє переходити на сайт з докладною інструкцією щодо користуванням програми.

Основне вікно *"Коефіцієнт теплопровідності композиту"* поділяється на два різних вікна, це модель Бургера та перколяції. Вікно вибору показано на рис. 2.2.

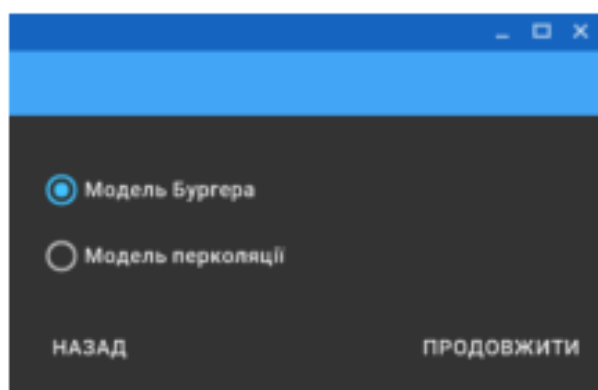


Рис. 2.2. Меню вибору моделей

Вікно *"Модель Бургера"* без параметрів та виводу результатів показано на рис. 2.3.

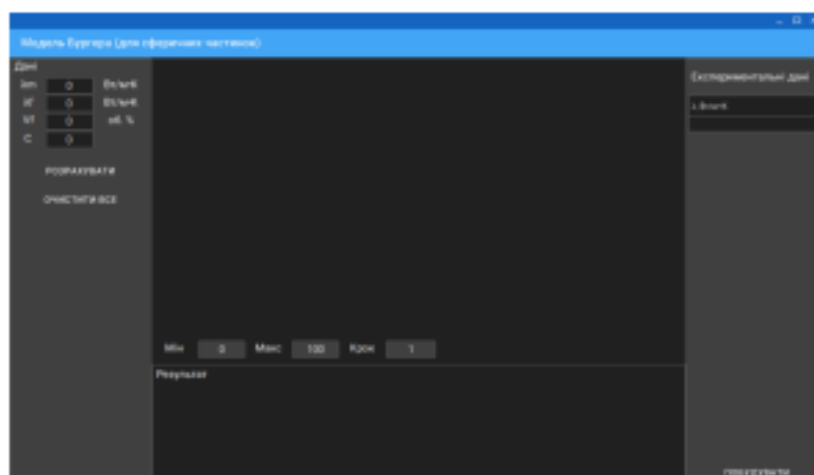


Рис. 2.3. Меню "Модель Бургера"

Вікно *"Модель Бургера"* з вказаними параметрами, розрахунками та виводом результатів у вигляді графіку показано на рис. 2.4.

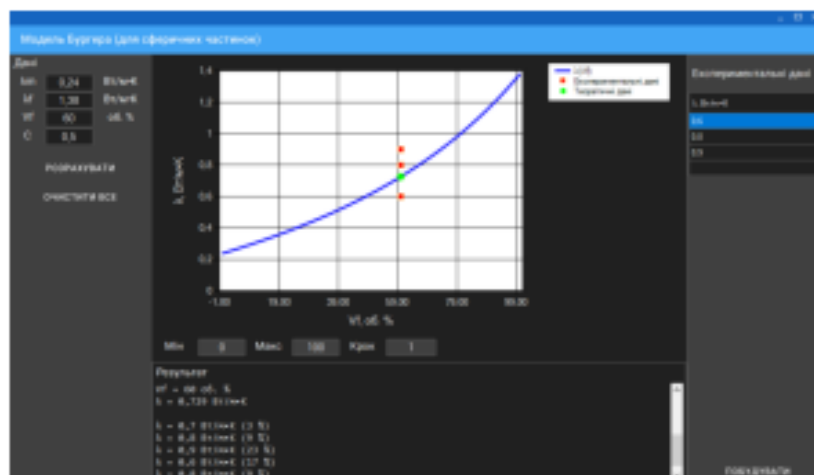
РОЗДІЛ 8.**АВТОМАТИЗАЦІЯ ТА ПРИЛАДОБУДУВАННЯ**

Рис. 2.4. Меню "Модель Бургера" з усіма розрахунками та виводом графіка

Розрахунок здійснюється за допомогою формули Бургера.

Графік будується за основними точками та за вказаним діапазоном.

Також на графіку виводяться теоретичні точки що розраховуються та експериментальні точки які вказує користувач.

Кнопка "ОЧИСТИТИ ВСЕ" видаляє початкові дані та очищує поля вводу та поле графіку.

Вікно "Модель перколяції" показано на рис. 2.5.

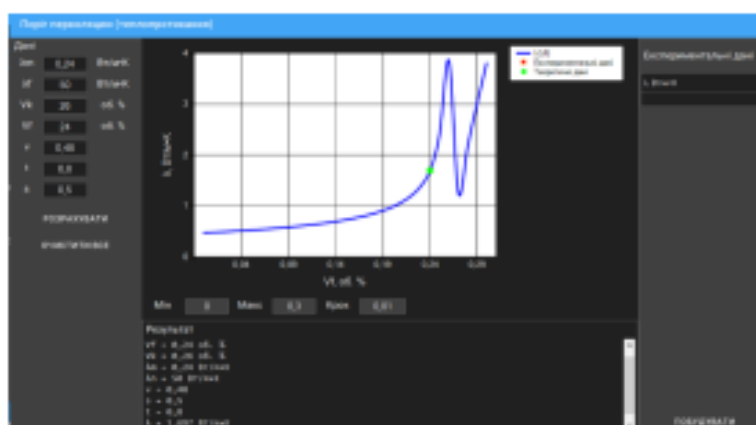


Рис. 2.5. Меню "Модель Перколяції" з усіма розрахунками та виводом графіка

Розрахунок виконується за допомогою класів.

Вікно розрахунку коефіцієнту "С" також складається з панелі вводу

даних, місцем виводу результатів розрахунку та графіка.
Вікно Коефіцієнт "С" показано на рисунку 2.6.

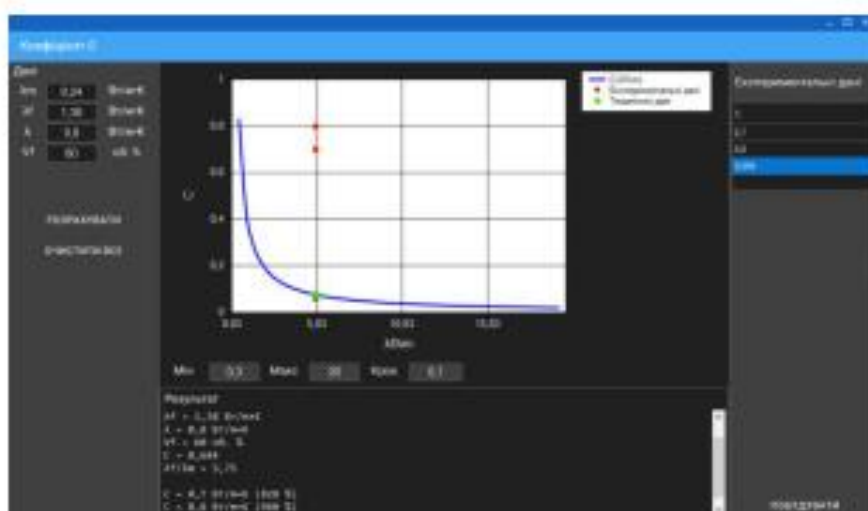


Рис. 2.6. Вікно Коефіцієнт "С"

Побудова графіків залежності коефіцієнта C від L/λ_m та перевірка на помилки здійснюються за циклом.

Вікно розрахунку об'ємної частки компонентів композитів показано на рисунку 2.7.



Рис. 2.7. Вікно "Об'ємна частка компонентів композиту"

Зберігання результатів доступне при натисканні правої клавіші мишки по полю з результатами. Є функції зберігання, копіювання та очистки.

Приклад вікна взаємодії показано на рисунку 2.8.

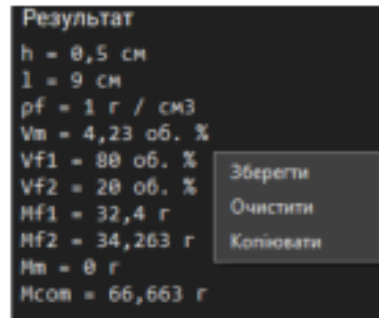
РОЗДІЛ 8.**АВТОМАТИЗАЦІЯ ТА ПРИЛАДОБУДУВАННЯ**

Рис. 2.8. Вікно взаємодії

Переваги програми включають в себе:

- розрахунок ефективного коефіцієнта теплопровідності: Програма дозволяє виконати розрахунки з використанням двох моделей, що сприяє зручному порівнянню результатів та допомагає визначити оптимальну модель для конкретного матеріалу;

- розрахунок та вибір коефіцієнта «С»: Програма допомагає визначити коефіцієнт «С» за експериментальними даними, що сприяє точнішому моделюванню теплопередачі в композитному матеріалі;

- розрахунок концентрації компонентів: Програма дозволяє користувачеві швидко розраховувати концентрацію компонентів, що спрощує процес розробки нових композитних матеріалів;

- порівняння експериментальних та теоретичних даних: Підтримка порівняння експериментальних та теоретичних даних дозволяє користувачеві отримати об'єктивний аналіз точності моделей та їх відповідність реальним умовам;

- представлення інформації мовами: Програма підтримує представлення інформації як українською, так і англійською мовами, що робить її доступною для широкого спектру користувачів.

Результатом є універсальний інструмент для вивчення та автоматизованого розрахунку поліімідних композитних матеріалів з високою ефективністю та зручністю використання.

3. Висновки

В результаті проведених досліджень і розробки програмного продукту "Теплопровідність поліімідного композиту ТсРС" нами було досягнуто важливих кроків у вивченні теплопровідності поліімідних композитних плівок. Описані теоретичні моделі, зокрема модель Бургера та модель перколяції, виявились ефективними у прогнозуванні теплопровідних властивостей матеріалу.

Програма розрахунку теплопровідності, створена на платформі NET framework та реалізована мовою програмування C#, виявилась потужним

інструментом для автоматизації розрахунків. Здійснено розрахунки ефективного коефіцієнта теплопровідності з використанням обох моделей, а результати виведено у вигляді графіків, що полегшує їх візуалізацію та порівняння.

Додатково, програма дозволяє розраховувати та вибирати коефіцієнт "С" на основі експериментальних даних, що забезпечує більш точне моделювання теплопередачі. Також надає можливість розрахунку концентрації компонентів та порівняння експериментальних та теоретичних даних для об'єктивного аналізу моделей.

Для моделювання комірок поліімідної композиційної теплопровідної плівки з рівномірним розташуванням частинок та автоматизування розрахунків ефективного теплопровідності композиту було застосовано програмний комплекс COMSOL MULTIPHYSICS.

Список використаних джерел:

- [1] Dylniev, G.M. *Thermal Conductivity of Mixtures and Composite Materials: Reference Book* / G.M. Dylniev, Yu.P. Zarichnyak. - L.: Energy, 1974. - 264 p.
- [2] Mikheev, V. A., Sulaberidze, V. Sh., Mushenko, V. D. Linear Model of Thermal Conductivity of Dispersed Materials Based on Polymer Binders // *Herald of Higher Education. Instrument Engineering*. 2017. Vol. 60, No. 3. P. 275-279.
- [3] Mikheev, V.A. *Ensuring the Quality of New Functional Materials for Heat-Conducting Coatings at the Development and Production Stages*. Ph.D. thesis in technical sciences. - 2018. 173 p.
- [4] Mikheev, V.A., Sulaberidze, V.Sh., Mushenko, V.D. Modeling the Thermal Conductivity of Three-Component Compositions // *Herald of Higher Education. Instrument Engineering*. 2016. Vol. 59, No. 7. P. 584-591.
- [5] Kirilov, V.N., Dubinker, Yu.B., et al. Thermal Conductivity of Silicon-Organic Elastomer Systems with Powdered Mineral Filler // *Ukr. J. Phys.* 1972. Vol. XXIII, No. 3.
- [6] Structural modeling and calculation of thermal conductivity of polyimide composite materials / V.M. Borshchov, O.M. Listratenko, M.A. Protsenko, I.T. Tymchuk, O.V. Kravchenko, O.V. Syddia, I.V. Borshchov, M.I. Slipchenko, // *Radiotekhnika: All-Ukr. Sci. Interdep. Mag.* 2022. №211. P. 133–142.
- [7] Patent RF No. 2600110 Heat-conducting electrically insulating composite material (variants) and method for its production, 2016.

ДОДАТОК В
Демонстраційний матеріал

